マイコン入門誌・マイコンボーイ

昭和58年5月1日発行

MICOMBOY

マイコン入門誌

マイコンボーイ

byモンキー・パンチ

創刊1号

530円



邪悪なコンピュータ・エイダの支配から人類を救え！

基本ベーシックで多くの機種に対応

掲載ゲームソフト

地雷原突破ゲーム《エレクトロマイン》

ギャンブルアドベンチャー《スロットボンバー》

フィナーレは、《レーザー・サブマリン》

Obunsha

3機種そろってエキサイティング。

左脳も右脳も興奮させる。

高性能のパソコンが手軽な価格で使いこなせる。—デビュー以来数ヵ月、FM-7に話題が集中しています。FM-8の先端技術を活かした豊富な機能と多彩なソフトウェア。数かずの魅力を携えて使う人を興奮させずにはおきません。

ご家庭のカラーテレビに接続して楽しめます。お手持ちのカラーテレビで美しいカラーグラフィック表示が楽しめます。市販のゲームソフト、プログラムの作成もOK。倍速モードも楽しめます。

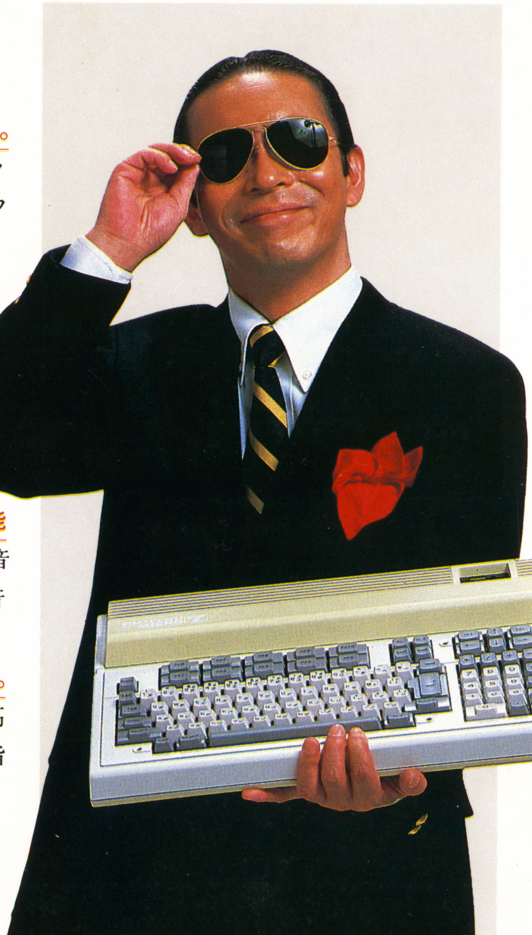
F-BASICを標準装備し、盛り沢山の機能をサポート。FMシリーズ用に機能強化したF-BASICを本体内に実装。プログラミングが容易で、FM-8用の流通ソフトのほとんどがそのまま使えます。

サウンドクルージングを楽しめるサウンド機能も内蔵。シンセサイザ用LSIを内蔵。三重和音までの音楽演奏が楽しめます。ゲーム効果音もバッチリ。外部スピーカの接続も可能です。

ドット毎に8色まで色指定できるグラフィック機能。カラーグラフィック画面は640×200ドットの高分解能表示が可能。1ドット毎に8色までの色指定ができ、パレット機能で色交換も簡単です。

読みやすく使いやすい日本語表示機能。オプションの漢字ROMカードを本体に装着すれば、漢字、英字、かな、カナ、記号など3,418種を表示、印字。日本語ワープロとしても使えます。

入門用から実務まで、1台でOKです。FMシリーズとして豊富に揃えた周辺装置により、目的に合わせてシステム構成できます。個人のデータ管理や、ビジネス用にも充分対応。



簡易言語を標準装備。

●操作は簡単、BASIC言語の知識も不要です。操作は質問応答方式。簡単な命令を指示するだけです。プログラミングをまったく必要としない対話形式になっていますので、BASIC言語を知らない人でも手軽に使いこなせるよう設計されています。

●身近な家計簿や住所録から、オフィスの各種資料まで幅広く利用でき、作表、計算、検索、分類、ファイリングなどの作業に威力を発揮します。これらの機能を駆使して最終的に表形式のファイルにまとめることにより、身近なものからオフィス内の資料まで十分に対応できます。

●大量のデータの入力が可能。画面やプリンタへの出力も自在にレイアウトできます。 ひとつの表について、20項目まで設定でき、最大250文字のデータ入力が可能です。しかも、FM-7のメモリ容量の限界まで入力できます。(1表250文字で55枚まで可能)。一度入力されたデータは、その内容や分類の仕方を問わず、簡単な検索条件と、画面やプリンタへの出力フォーマットを指定するだけで、必要な資料を短時間に作成できます。



先端技術が夢中にさせる興奮パソコン。**新発売**

FM-7
セブン

¥126,000(本体価格 簡易言語ソフト付)

パソコンとワープロが使いこなせるテレビ新番組

富士通提供「コンピュートないと」好評放映中

●放送日：毎月曜日、夜11時15分〜11時45分 ●放送局：テレビ東京/テレビ大阪

●出演者：司会/相原友子：講師/石田晴久(東京大学教授)他に有名ゲスト、レポーターが登場

※テキスト「パソコングラフィックス入門」工学社より発売中(定価650円)

——信頼と創造の富士通

富士通のパソコンFMシリーズ。

ディスク搭載、16ビットが興奮させた。

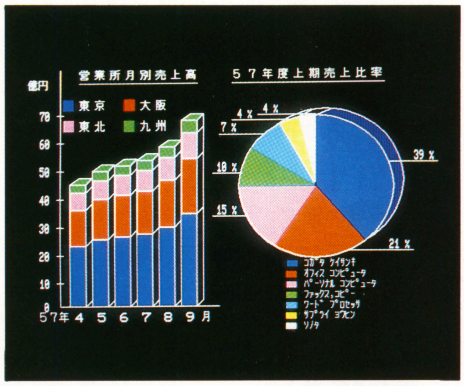
オフコンにせまる本格派。EDP部門から現場まで本格的ビジネスユースに対応できます。

FMシリーズの高級機種として、FM-11が新登場。EXタイプはCP/M-86を標準装備。これによりCIS-COBOL(ANSI-74準拠)、Pascalなどの高水準言語を利用できます。そして、BASICは、使いなれたFM-8と上位互換性のあるF-BASIC。オフコンでのデータ処理を分散化したいというEDP部門と、今まで使っていたFM-8のBASICでデータ処理したいという現場双方で、同一機器の使用が可能になりました。

フロッピィディスクを標準装備、そして分離型。操作性、扱いやすさをさらに向上。 FM-11は、本体、キーボード分離のセパレート・タイプ。本体とキーボードとの接続コネクタは、本体裏側から。しかもケーブル(カールコード)は2.5mの長さです。キーボードは操作性を重視したロープロファイル、シリンドリカル・ステップ・スカルプチャー設計。FM-11EX、ADの本体にはフロッピィディスクが1ドライブ標準装備され、さらに1ドライブ増設可能です。

日本語処理機能をいっそう充実。 日本語処理をサポートしたCP/M-86の採用により、COBOLなどの高水準言語でも日本語処理が行なえます。もちろんF-BASICでも処理可能。カーソル位置に漢字表示でき、漢字の拡大、外字登録も可能、表示文字数は最大40字×25行、などなど、使いやすさは抜群です。(漢字ROMカード、オプション)

メモリは128KBを標準実装、最大1MBまで増設可能。 メインメモリとして、128KB標準実装されており、さらに本体内の拡張スロットに増設RAMを実装すれば最大1MBまで拡張できます。これによりCP/M-86、MS-DOS、OS-9などのOSのもとで、大容量メモリを必要とする高水準言語を余裕をもって利用できます。



多彩な機能を発揮するCRT表示。 最大640×400ドットのグラフィック表示は、キャラクタとの混在表示も可能。カラー表示は16色。最大12画面を瞬時に切り換えて表示できるマルチページ機能、色交換が簡単にできるパレット機能、上下さらに左右へも可能になったスクロール機能、などCRT表示機能がより多彩になりました。

簡易言語「FMCALC」を標準装備。

●コマンドは18種類。うち10種類がファンクションキーに設定されています。「FMCALC」は、FM-11の電源を入れると同時に実行され、画面に表が現われます。わずか18種類のコマンドで操作でき、そのうち10種類がファンクションキーに設定されているので、ワンタッチ操作ができ、さらに使い易くなりました。

●目的に合わせた処理ができ、会計業務、帳簿作成、統計処理など本格的ビジネス分野に利用できます。「FMCALC」は、プログラミングを知らない人でも「FM-11」を使いこなせるビジネスソフト。合計値や計算式など縦横のすべての関連数値を、データがひとつひとつ変わるごとに瞬時に計算します。表はスクロールによって、横52項目×縦128行まで作成できます。

ニーズに合わせてお求めやすい3タイプ。

●EXは、薄型フロッピィ1ドライブを標準装備。CP/M-86、F-BASIC、FMCALCを添付。

●ADは、薄型フロッピィ1ドライブを標準装備。F-BASIC、FMCALCを添付。

●STは、F-BASIC(ROMバージョン)。

*FLEX、UCSD Pascal、OS-9、CP/M-80・CP/M-86、MS-DOSはそれぞれTSC社、カリフォルニア大学理事会、マイクロウェア社、ディジタル・リサーチ社、マイクロソフト社の登録商標です。



ビジネス用途を大きく拡げる高級パソコン。**新発売**

FM-11

イレブン

■FM-11 EX ¥398,000(本体価格 簡易言語ソフト付)
■FM-11 AD ¥338,000(本体価格 簡易言語ソフト付)
■FM-11 ST ¥268,000(本体価格)

高級ホビーからビジネスまでの多才パソコン。

FM-8

エイト

¥218,000(本体価格)

富士通

富士通株式会社 半導体統轄営業部 〒105 東京都港区虎ノ門2-3-13 TEL.(03)502-0161
●札幌営業所(011)271-4311 ●東北営業所(0222)64-2131 ●長野営業所(0262)26-8222 ●静岡営業所(0542)54-9131
●名古屋営業所(052)201-8611 ●大阪営業所(06)344-1101 ●広島営業所(082)221-2288 ●九州営業所(092)411-6311
マイコンスカイラブ●東京・虎ノ門(03)591-1091/591-2561 ●東京・秋葉原(03)251-1448/251-1449 ●大阪(06)344-7628/341-0486
●名古屋(052)221-6016 ●札幌(011)241-4185 ●広島(082)247-3949

'83全国高校生 パソコンソフトコンテスト

知的好奇心や旺盛な探究心は，若者の特権なのだ。その特権を十分に生かせば，パソコンだって使いこなしは自由自在。
才能が光れば，名誉も栄光も，そして富までもころがりこむのだ。

旺文社主催の，'83 全国パソコン　ソフト　コンテストは，グランプリに，なんと30万円の賞金。さらに富士通のパソコン，FM－7フルセットが副賞だ。コンテストにチャレンジして，若さを証明しよう。

'83全国高校生 パソコン ソフト コンテスト 応募要領

●**応募資格**　全国の高校生（個人ならびにグループ）

●**応募作品**　FMシリーズで動作するプログラムで，F－BASICあるいはマシン語で作成したプログラム。フロッピィディスク（5インチ，8インチいずれか），カセットテープで参加。（必ずプログラムリストを添付の上，氏名明記）応募ソフトはゲーム・学習用などなんでも可。ただし，未発表のオリジナル作品に限る。全ての応募作品は返却しません。

●**送り先**　〒162 東京都新宿区横寺町55　旺文社「全国高校生パソコン ソフト コンテスト」事務局宛

●**締め切り**　昭和58年9月30日（当日消印有効）

●**審査員**　全国高等学校校長会会長・増田信　東京大学教授・石田晴久，タモリ，富士通㈱専務取締役・安福眞民，旺文社雑誌編集局長・石川卓司

●**入賞者・入賞作品の発表**　昭和58年12月14日（旺文社発行高校誌・昭和59年1月号誌上）

●**応募方法**　作品に必ず応募票（見本参照）をつけて郵送で応募。

●**入賞作品の著作権**　著作権その他一切の権利は旺文社ならびに富士通に帰属する。作品の商品化の際には，著作者と協議のうえ対価をお支払いいたします。

★各賞と賞品★

グランプリ……………………………1　点
奨励金30万円，FM－7一式（本体：MB25010，カラーCRTディスプレイ：MB27303，カラーCRT用ケーブル：MB26512，薄形ミニFDユニット：MB27607，FM－7用ミニFDインタフェースカード：MB22407，ドットプリンタ：MB27404，プリンタ用ケーブル：MB26514），表彰状・楯，旺文社宇宙科学シリーズ　〈学校賞〉旺文社宇宙科学シリーズ，表彰状・楯

優秀賞……………………………2　点
奨励金10万円，FM－7（本体：MB25010）または薄形ミニFDユニット：MB27607，FM－7用ミニFDインタフェースカード：MB22407，表彰状・楯・旺文社宇宙科学シリーズ　〈学校賞〉旺文社宇宙科学シリーズ，表彰状・楯

特　選……………………………3　点
奨励金5万円，FM－7用漢字ROMカード：MB22405，ソフト10本，表彰状・楯，旺文社宇宙科学シリーズ　〈学校賞〉旺文社宇宙科学シリーズ，表彰状・楯

佳　作……………………………10　点
図書券（5000円），ソフト5本，万年筆，表彰状・楯，旺文社宇宙科学シリーズ　〈学校賞〉旺文社宇宙科学シリーズ，表彰状・楯

（パソコン ソフト コンテスト応募見本）

作品の題名							
作品の内容							
クラブ名（グループ名）		人	先生の氏名		㊞	教科名	
代表者名（個人名）		男・女	学校名		学校TEL	学年	

FMシリーズ100台，モニター募集中。利用しないって手はないぞ

パソコンショップで腕はきたえているけれど，パソコンないからコンテストに参加できない，とあきらめるのは早計。コンテストにむけて，協賛スポンサー富士通が，FMシリーズ（本体）100台のモニターを募集中なのだ。資格は，高校生であることと，パソコンクラブ，あるいは何人かのグループであることの2点。仲間を集めてグループを作れば，だれだって申し込めるのだ。

チャンスは自分の手でつかむもの。指をくわえて見てるだけなんてのは，若者じゃないぞ!!

★

●**モニター資格**
高等学校に在籍する生徒で構成された"グループ"及び"パソコン・クラブ"。両者とも顧問の先生，または指導の先生（1名）が必要。

●**応募方法**
・規定の応募用紙に，必要事項をもれなく記入し「全国高校生パソコン ソフト コンテスト」事務局まで申し込む。
・クラブ，グループの活動方針，現在の活動状況，過去のソフト関係実績等を明記し，モニターとして何をやりたいか等の点について原稿用紙（400字）3枚以内にまとめ，応募用紙とともに事務局まで。

●**応募締切り**　昭和58年6月25日　事務局必着

●**選考方法と通知**
事務局で書類選考の上，顧問先生，代表者に通知，同時にモニター機を7月上旬に発送。

●**モニター規約**
・モニター機は，富士通FMシリーズ本体（FM－7，FM－8，FM－11のいづれか）。尚，機種に関しては主催者側で決定させていただきます。
・モニターに選考されたクラブ，グループは，「全国高校生パソコン ソフト コンテスト」に作品を必ず応募することが義務づけられる。

（モニター申し込み応募用紙見本）

富士通FMシリーズモニターに応募します。
クラブ・グループ名（クラブかグループかを明記）人数
学校名 住　所 Tel
顧問教師名・担当教科名　㊞
生徒代表者名・学年・クラス
現在使用しているパソコン（メーカー名，機種名）

パソコン番組は，これからのTVの主役なのだ



司会の石田晴久教授と相原友子さん。

ついにというか，満を持してというか，パソコン番組のTV放映がスタートした。4月4日からはじまった，テレビ東京，テレビ大阪の〝コンピュートないと〟は，司会にコンピュータの権威，石田晴久東大教授と相原友子。　コンピュータグラフィックスから最新パソコン情報まで，毎週月曜日の夜11時15分から30分間，まさにコンピューター色なのだ。もちろんパソコン大好き少年にもパソコン入門のキミにも，気軽に見て楽しめるバラエティ番組だ。番組の中で，パソコンのソフトについても，あれこれ紹介してくれるので，コンテストに参加するつもりなら見のがせないぞ。


マイコンボーイ MICOMBOY
●創刊記念スペシャル・ストーリー
RUNだ!! 乱

電子地雷エレクトロマイン

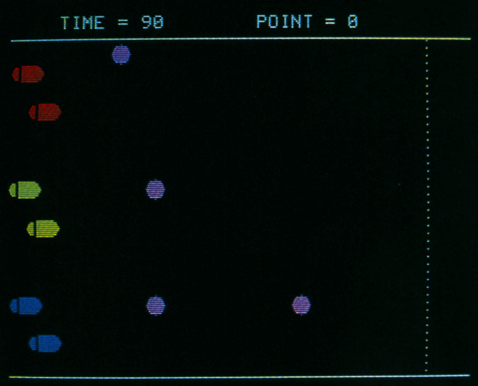


時は2119年。世界の秩序機構を管理する超巨大コンピュータ・エイダが突如暴走し，凶器へと変身しはじめた。

しかも，コンピュータの中枢へつながるゲートは次々とエイダ自らの手で閉ざされつつある。そして，この修復のために内部に侵入する者は次々に殺された。今や刻一刻と人類は邪悪なコンピュータの支配下におかれつつあるのだ……。

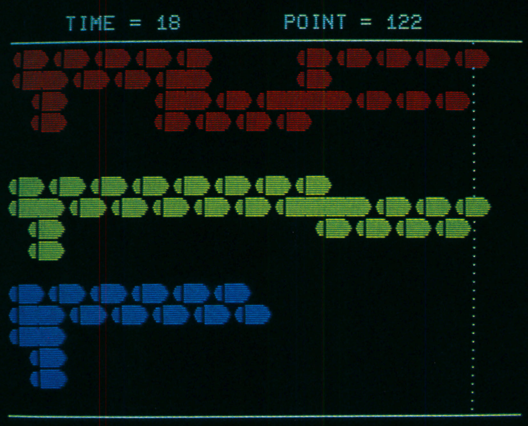
そこに，一人の少年が立ち上がった。その名は乱!

その乱を待ち受けるエイダ第一の攻撃は電子地雷。暴走のために，エイダ内部の電子ベルトが危険な地雷原と化してしまったのだ。エイダの中枢へつながる電子地雷原を果たして乱は突破できるか!?

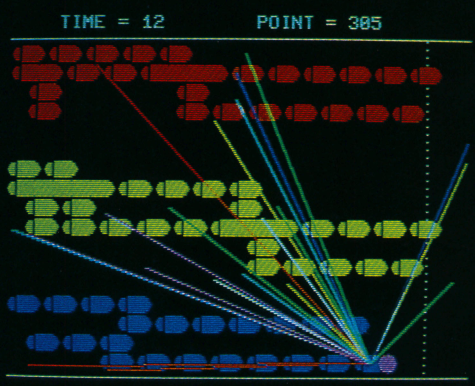


▲地雷は一瞬の間だけ見ることができる。後は記憶を頼りに地雷をさけながら歩いてゆかねばならない。





◀記憶を頼りに一歩一歩進んでいく。三人がゴールに到着すると、さらに地雷がふえて新たな地雷原へ突入!!



▲うっかり地雷を踏んでしまうと，たちまち爆発を起こしてGAME OVER!!

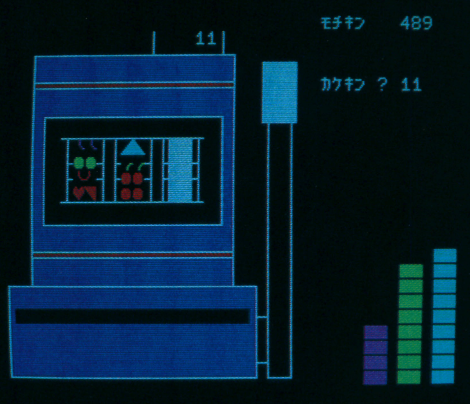
 FROM P.77

スロットボンバー



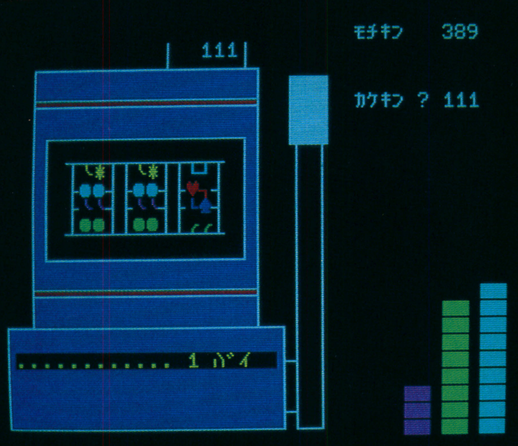
超巨大コンピュータ・エイダが暴走した。そのエイダを破壊すべく，内部に進入した乱。電子地雷エレクトロマインの危険を逃れた乱を待ちうける第二のアドベンチャーは意外にもスロットマシン。

しかし，このスロットマシンもエイダの手であやつられ，危険なシロモノに変わっていた。コンピュータ・エイダの中枢に到るには，どうしてもスロットマシンで，情報コインを得なければならない。このスロットマシンは，乱数を駆使しており，出る絵柄の行方は神のみが知っている。手元にあるコインは500。これをもとに乱はどこまでコインを増やすことができるだろうか？

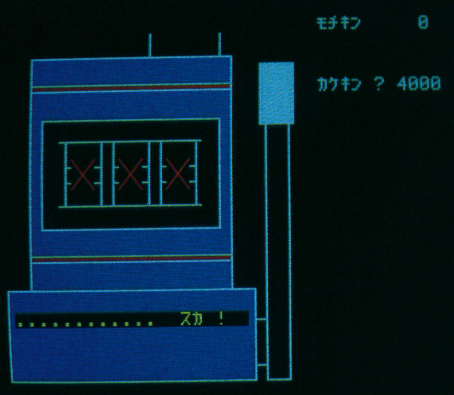


▲ゲームのルールは簡単だ。持っているコインの範囲で賭ける。あとはスペース・バーを押すだけ。



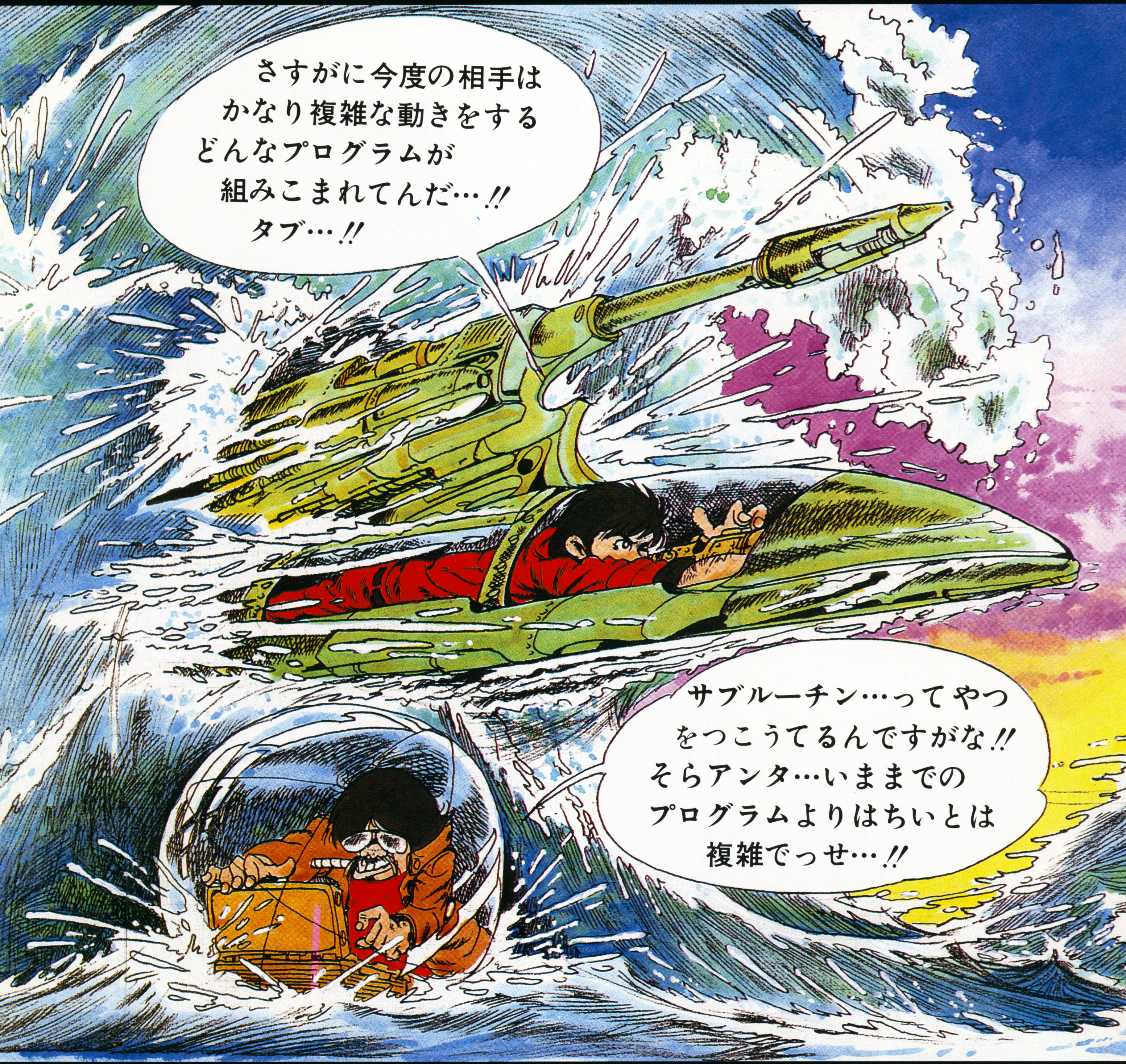


◀スペースバーを押すたびに絵が止まる。しかし、すぐにではない。乱数をもとにいくらか余分に回った後で止まるのだ!



▲コインをすべて失って負けると，スロットマシーンは×印を出し，消えてしまう。

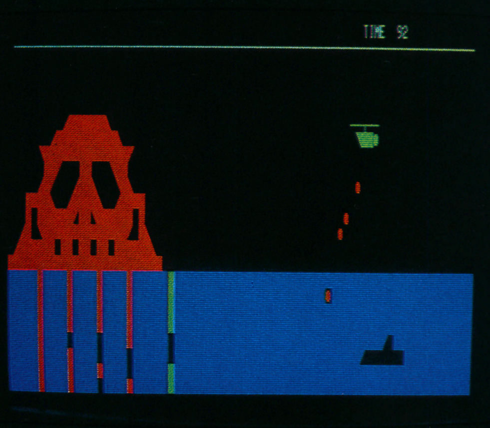
レーザー・サブマリン



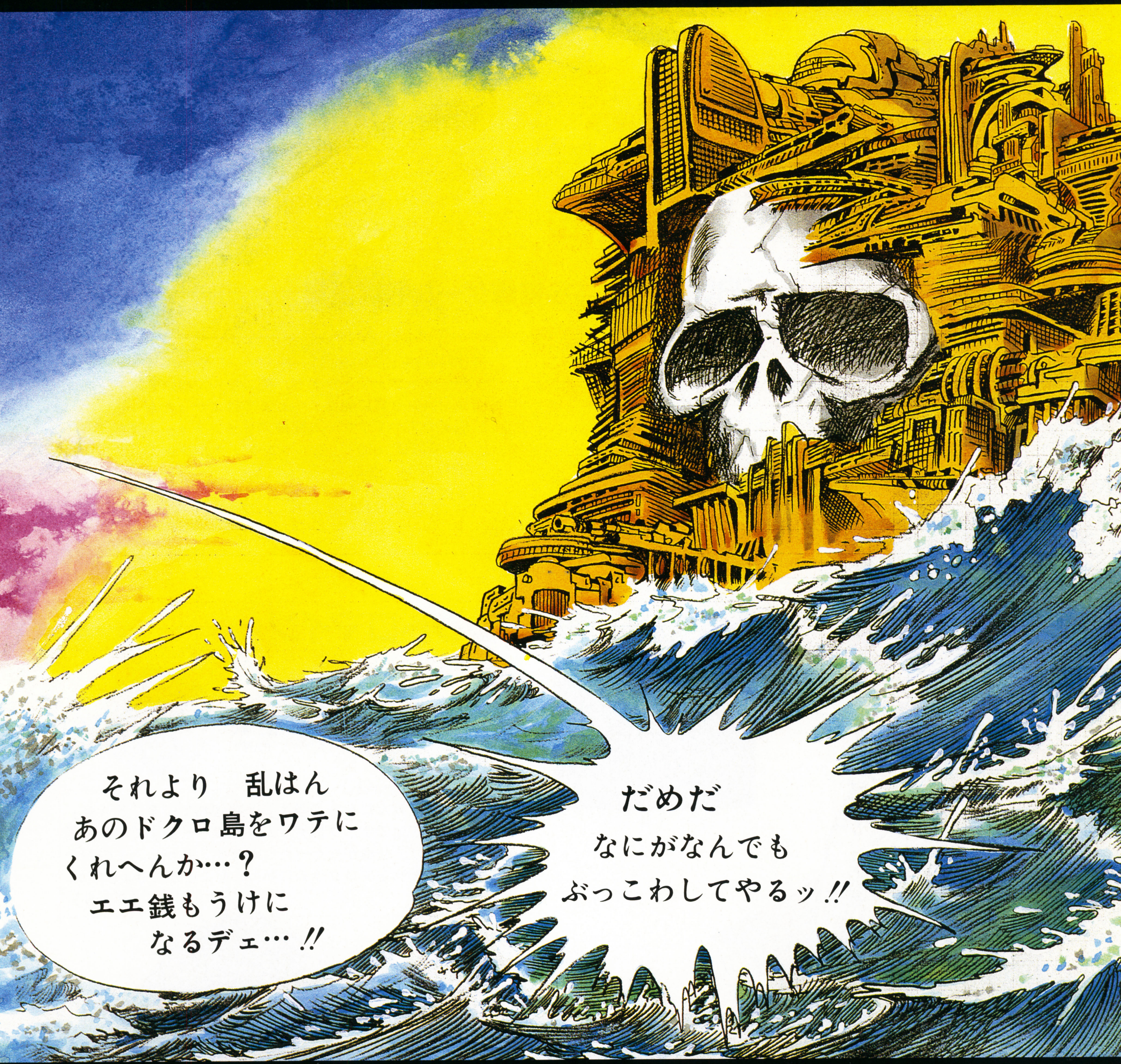
人類を支配する邪悪なコンピュータと化したエイダ。エイダは，そのエネルギーを地底のマグマ熱から得ている。エイダを破壊するには，海中にのびている４本のヒートエネルギーパイプを切断するしかない。

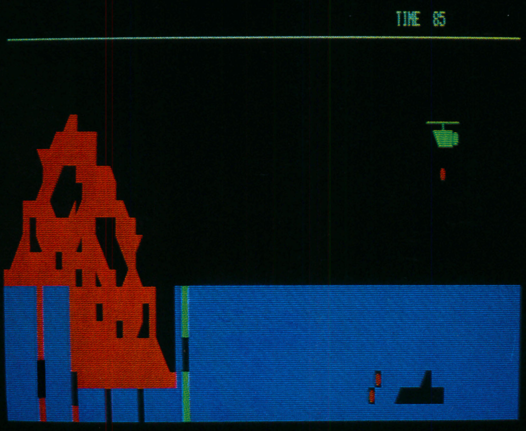
しかし，海上において対潜ヘリが絶えず爆雷を投下し続けている。しかも，海中にも防御シールドがめぐらされている。

弱点はただひとつ。防御シールドにあいた，小さなすき間だけだ。ここからレーザー砲を打ち込むのだ。しかし，レーザー砲でエネルギーパイプを切断しても，絶えずパイプはのびてくる。「RUNだ!!乱」のクライマックス！ 乱は潜水制限時間内にエイダを破壊できるか？

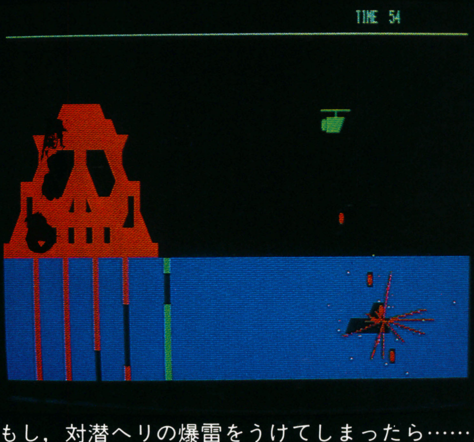


▲対潜ヘリの爆雷攻撃をさけながら，レーザー砲で４本のエネルギーパイプを切断することが使命だ。





◀みごとにエネルギーパイプを4本とも切断できれば、ドクロ顔をしたコンピュータ・エイダは海中に崩壊していく。



▲もし，対潜ヘリの爆雷をうけてしまったら……。潜水艦は大爆発を起こして THE END。

目次 ©旺文社／1983

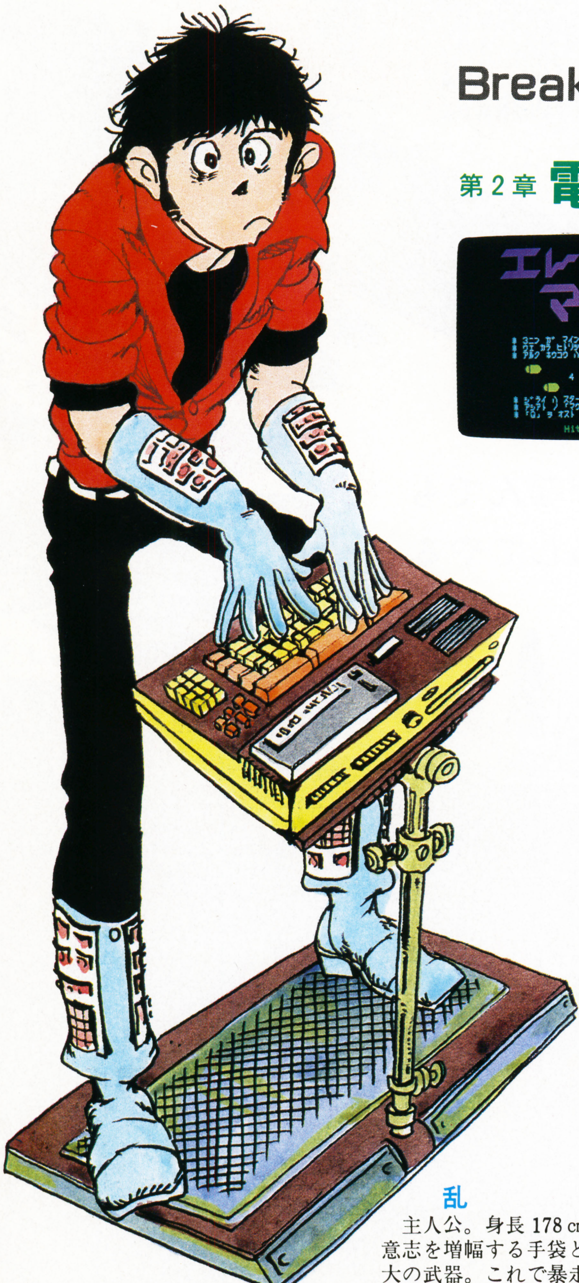
序

MI COMBOY（マイコンボーイ）は，マイコンに興味のある人，習得したいと思ってもメカには弱いからとあきらめている人，必要にせまられているが，よくわかる入門誌がみつからなくて困っている人――そうした人たちの強い要望に応じるために創刊しました。

だから，《楽しく学べること》，《やさしく学べること》のために，マンガ構成によるゲーム・ストーリーとしました。ごらんのように，あの有名な《ルパン３世》の生みの親であるモンキー・パンチのスマートなマンガのストーリーを楽しんでいく内に，ごく自然にベーシックなマイコンのプログラムがわかるようになります。

また，マイコンを持っていない読者のためにも，ゲーム

（From Editor Z68086）
ベーシック・ワールドへ，モンキー・パンチのイントロダクションと共に発進！　とまではよかったが……。
編集のすべてがゲーム同様アドベンチャーの連続で，
　0　GOTO　999：' クルシンデ　クルシンデ　クルシム
　999　REM　HELL
すなわち，Go to hell！の構文でした。



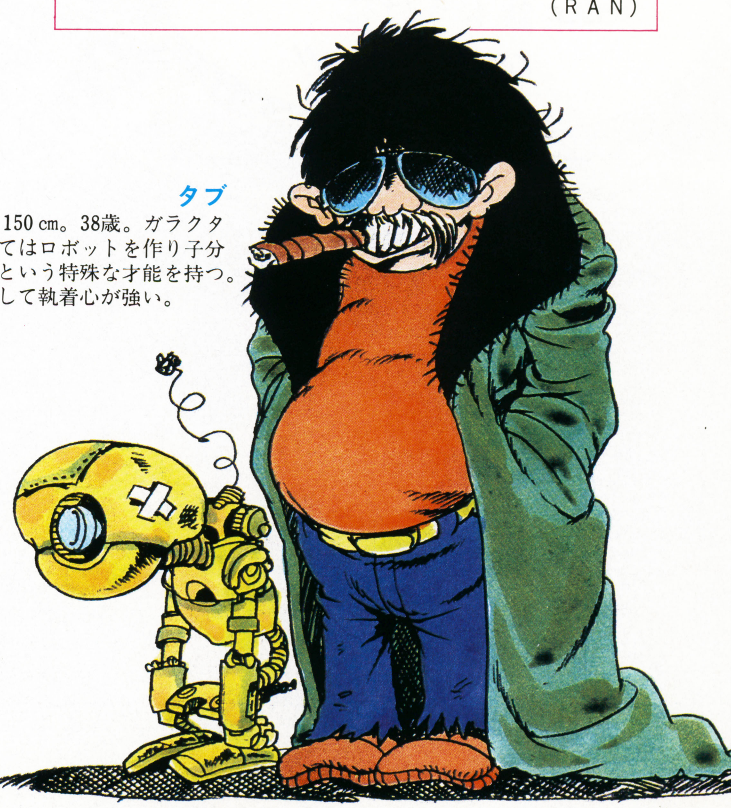


解説
成瀬潤行

ゲーム・プログラマー
佐武正敏
パスワード「RAN」



（From Programer）
なぜか，みんなから「酒乱」と呼ばれている「ＲＡＮ」こと佐武正敏です。プログラムを完成させるまでに最も困ったことは，仕様がなかったことです。しかし，そのおかげで私の個性をプログラムに折りこむことができましたが。
（ＲＡＮ）



タブ
身長150cm。38歳。ガラクタを集めてはロボットを作り子分にするという特殊な才能を持つ。金に対して執着心が強い。

乱
主人公。身長178cm。18歳。意志を増幅する手袋とくつが最大の武器。これで暴走コンピュータ・エイダに立ち向かう。

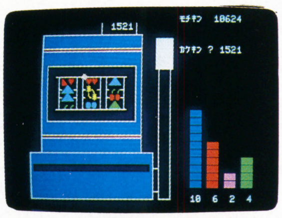
表紙AD＝八木　保

の説明などでディスプレイの表示を効果的に出し，充分楽しめるようにしてあります。当然のことながら，マイコンを持っている人は，本誌のプログラムでゲームを楽しむことができます。しかも，市販されている多くの機種で楽しめます。いわば，読者の皆さんとマイコンとを結びつける仲介役こそが，マイコンボーイの役割というわけです。

マンガ／**モンキー・パンチ**



ランダム
身長181cm。正体不明の，沈着冷静なアンドロイド。



解説
天道襄治

ゲーム・プログラマー
佐竹英樹
パスワード「Nor」

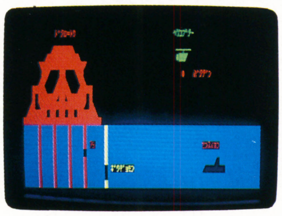


(From Programer)

真夜中，ペンを口にくわえ，ズボンのポケットに手を突っ込んで狭い部屋の中をうろつき歩きまわる。ときどき，リストになぐり書きをする。コーヒーを飲み，「あれえ，おかしーなあ」とぼやきながら，キーボードをたたく。

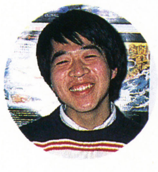
スロトマシンは，こうして製作されたのである。ああ，なんと知的な作業であろうか……。　（Ｎｏｒ）

（ほ～んとに知的な仕事でしたわね？　カゲのプログラマー・Lei）



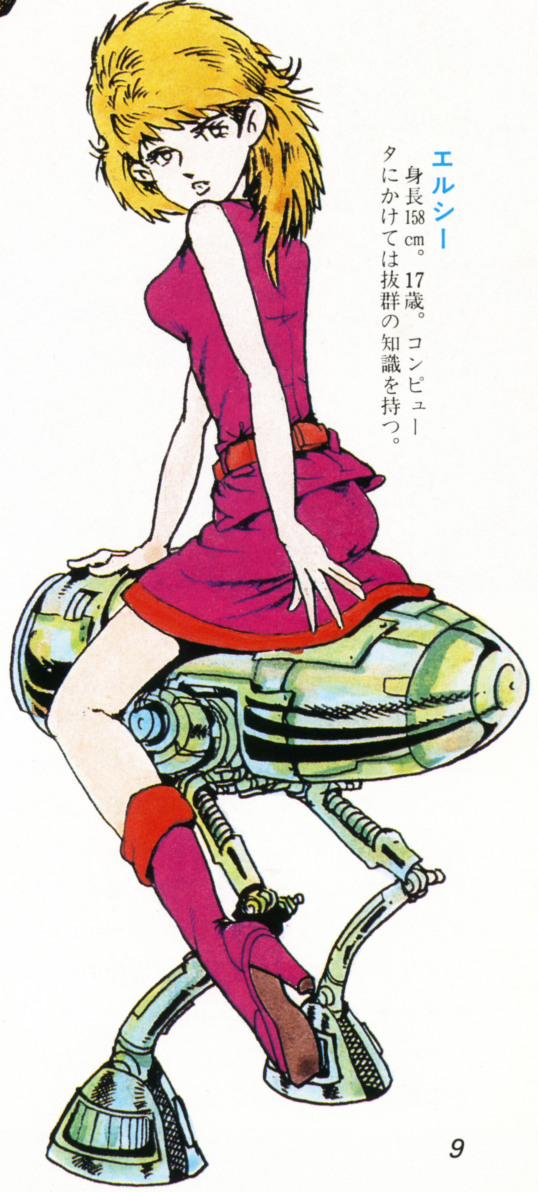
解説
相馬レイ子

ゲーム・プログラマー
太田　淳
パスワード「ETX」



(From Programer)

ＵＮＩＶ.＝電気通信大学（もっと有名にしたいナァ）：性格＝（二重人格 ＡＮＤ 両極端）？：血液型＝ＡＢ：星座＝双子座：ＩＦ　ＹＯＵ＝マイコンネクラ族　ＴＨＥＮ ＰＲＩＮＴ〝マイコンに没頭しすぎ！ もっと自分の身のまわりのいろんなものに興味をもとう〞（ＥＴＸ）



エルシー
身長158cm。17歳。コンピュータにかけては抜群の知識を持つ。

次号は7月8日発売予定！

RUNだ!! 乱

第1章 マイコン遭遇



マイコンボーイ創刊記念スペシャルストーリー「RUNだ!! 乱」は，マイコン入門のために３本のゲームソフトとマッチさせた物語で，舞台は西暦2119年だ。この上の物語と下の解説をあわせて見ていけば，キミも物語を楽しみながらマイコンのさわり方から実際のゲームプログラムのつくり方まで身につけることができるだろう。

まず最初は，マイコンに関するＡＢＣとベーシックによるプログラムの入門だ。キーボードのさわり方から，実際のプログラムの基本まで，主人公の乱といっしょに，アドベンチャーを楽しみながら見ていこう。そして，プログラムのルールなどを覚えたところで，いよいよマイコンゲームの世界へチャレンジだ！

第1種マイコン遭遇 PART1 コンピュータとの出会い

コンピュータがはじめてつくられたのは1945年だから，物語の舞台2119年からすると 174 年前，２世紀近くも昔のことになる。現在の加速度的な進歩からみて，2016年製造のコンピュータは，主人公の乱にとってかなりの年代物といえる。

さて，コンピュータという言葉を知らない人はいないと思うけれど，「いったいどんなものなの？」との質問に，キミならどう答える？

「なんでもできるキカイ」……まあそうだ。しかし，これにはタダシ書きがつく。「人が教えてやれば……」と。驚く



なかれ，スイッチオンしただけのコンピュータには1＋1のようなたし算すらできやしない。「ウソでしょ？」

ホントなのだ。コンピュータにわかる言葉で命令してやる必要があるんだ。(この場合はベーシックでPRINT1＋1と命令する)

この命令は，上で２人がながめているキーボードから打ち込む。キーボードやディスプレイなど，コンピュータ本体をハードウェア*というのに対し，命令のことをソフトウェア*という。いくら本体が立派でも，**ソフトが無ければタダの箱というワケだ**。

今までのキカイには，こんなヤッカイな命令なんていらなかった。テレビはスイッチオンで番組を見せてくれる。しかし，コンピュータはそうはいかない。少しメンドウだ。

でも，メンドクサク思える命令こそ実はとっても便利なシロモノなんだ。命令が必要だからこそ〝なんでもできるキカイ〟なのだから。

テレビは番組しか見せてくれない。その点コンピュータは命令しだいで，計算はもちろん絵も描くし，音楽の演奏もする。それこそ，なあ〜んでもしてくれるんだ。

コンピュータは入力装置であるキーボードから命令を受けとり，内部で実行し，ディスプレイ*その他の出力装置で実行した結果をキミに報告する。敏速かつ正確無比だ。

上の２人は，まさにコンピュータと人間の会話の接点であるキーボードの上に乗っているワケだ。

＊ ハードウェア＝金物という意味で，心に対する体のようなものだ。
＊ ソフトウェア＝ハードウェアに対しての造語だ。
＊ ディスプレイ＝文字や記号を表示するＴＶ画面のことだ。



第1種マイコン遭遇 PART II マイコンに命令せよ

ダイレクト・モード*って何だ？

マイコンと一問一答のかたちでやりとりするのがダイレクト・モードだ。つまり，マイコンにさせたい仕事をひとつだけ命令して，その場で実行してもらう命令だ。電卓と比較しながらダイレクト・モードの例をあげよう。

電卓　5 × 3 = → 表示 15

マイコン　P R I N T* 5 * 3 RETURN（リターン・キー） → 表示 15

電卓とよく似てるだろ？　単純だけどとても大事だ。これがすべての基本になる命令パターンなんだから。

プログラム*・モードって何だ？

マイコンにやらせたい仕事とその順番を前もって命令しておいて，あとでまとめて実行してもらおうっていうのが，このプログラム・モード。つまり，前のダイレクト・モードをひとつひとつ順序よく積み重ねて，いっぺんにやろうというわけだ。

この命令には，フォートラン，コボル，**ベーシック**などいろんな言葉がある。ここでキミたちにマスターしてもらうのは，マイコンでいちばん使われているベーシックだ。その名の通り基本的な言葉だから，あせらず落ち着いて取り組んでいこう。右頁の例を見ると，ダイレクト・モードが基本になっていることがわかるだろ？

＊ ダイレクト＝直接，モード＝方式
＊ PRINT （プリント）＝ディスプレイへの出力をさせる命令。
＊ プログラム＝予定・計画などの意味がある。
＊ RETURN＝もどるという意味があり，このキーで命令はマイコンに伝わる。



命令を，仕事の手順通り順序よくマイコンに伝えるためには，何か目印が必要だ。そのためにベーシックで使われているのが**行番号**とよばれるものだ。

マイコンはこの行番号を，各命令文の目印として理解し，プログラム実行のときの手順の目安にする。

あとで命令を書きたす場合のことを考えて，10番おきにつけておくとよいだろう。便利だよ。

簡単なプログラムの例を示そう。

ここでは「RUN ダ ラン」という文字をディスプレイに表示するという仕事をさせている。

右上のようにプログラムを書き出したものを**プログラムリスト***というんだ。

行番号　　命令の部分

```
10  A$="RUN  ダ   ラン"
```
(A$という箱にRUN　ダ　ランという文字を入れろ！)

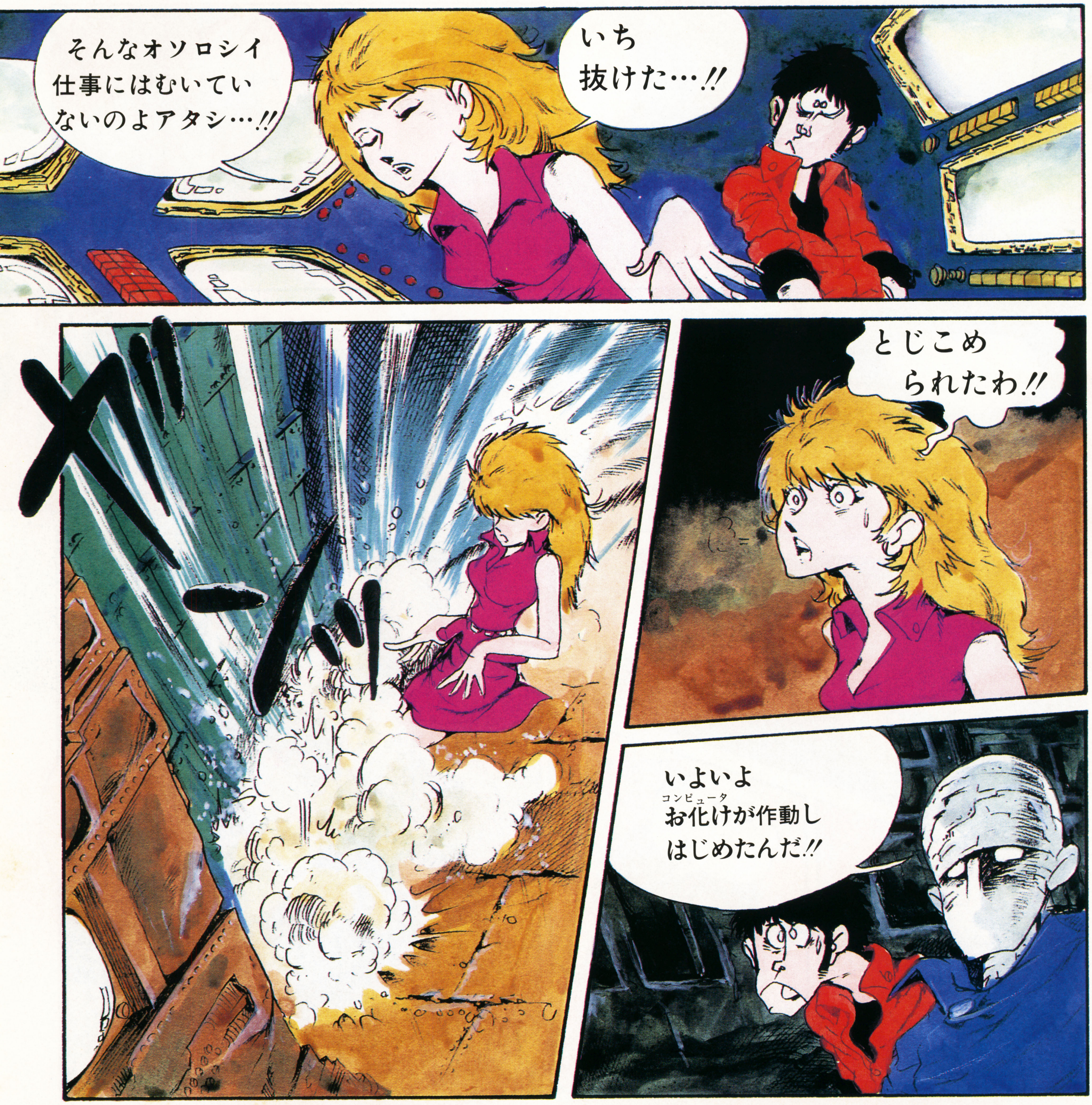
```
20  PRINT  A$
```
(A$という箱の中味を表示しろ！)

```
30  END
```
(仕事は終わり！ゴクロウサン……)

ひとつの流れを持った仕事を，簡単なダイレクト・モードのような命令にキザンである。

頼もしいようでもそこはキカイの悲しさ。仕事の説明は手とり足とりしてやらないと動けない。

＊　リスト＝一覧表のことだ。



第2種マイコン遭遇 PART I NEW,LIST,RUN

ベーシックの命令の中には，プログラムの入力や実行に関係するものがある。

NEW，LIST，RUNの３つだ。

特にRUN命令は，これがないと，せっかく打ち込んだプログラムが仕事をしてくれないという大事な命令だ。

どの命令も，RETURNというキーを押してはじめてマイコンに伝わるようになっている。RETURNキーは，キーボードの中でいちばんよく使うキーといえるね。

さて，次はそれらの命令の具体的な使い方の話だ。第2種マイコン遭遇といこう。

NEW ： ニュウ／新しいという意味の英語だね

N E W → RETURN と押す

《役目》 メモリの内部にすでにあるプログラムを消去して，変数*をカラにする。字を書く前に黒板をきれいにするのと同じ。

新しいプログラムを実行する前に行う命令だ。

これをしないと，直前に入力したプログラムが，新しいプログラムに混入するなどのトラブルが起こりがちだ。

行番号はふつう10番おきにつけるけど，たまたま以前のプログラムに25番なんてのがあったりすると，新しいプログラムの中にそのまま残ってしまうからだ。

用心するに越したことはない。ぜひやっておきたい。

＊ 変数＝16頁で説明している。



LIST ： リスト／もとの意味は一覧表

L I S T → RETURN と押す

《役目》 メモリ内部のプログラムをディスプレイに表示させる。

行番号10，20，30……と入力していって，10番と20番の間に何か入れ忘れていたことに気付いたとする。そうしたら，行番号15として命令を入力すればいいのだけれど，ディスプレイには行番号の若い順には並んでいないね。

ここでLIST命令をすると，ディスプレイに，行番号の若い順に整理されたリストが表示される。

マイコンの中に，今どんなプログラムがあるのか知りたいときにもLIST命令を使えばOKだ！

RUN ： ラン／走る！だ

R U N → RETURN と押す

《役目》 作業開始。

この命令で，メモリ内部のプログラムの実行がはじまる。

マイコンは命令された通りにしか動かない。

キミの希望するのと違った動き方をするなら，それはキミの入力したプログラムに問題があるからなんだ。こんなときは左のLIST命令でプログラムリストを表示させ，スクリーン・エディット*で改善していこう。

さて，乱のRUN命令で，巨大なお化けはいったいどんな実行をはじめるのだろうか？

＊ スクリーン・エディット＝ディスプレイ上で入力したものの訂正を行うことだ。



第2種マイコン遭遇 PART II INPUT

いよいよ実際のプログラミングだ。

RUNは作業開始，プログラムの実行を開始させる命令だったね。乱がRUN→RETURNとキー操作したら，カワイコチャンが「指令コードを入れなさい」とコンピュータからの指令を伝えてきた。まず，このようなことをマイコンでやらせてみよう。こういう場合, INPUTという便利な命令があることを覚えておこう。

INPUT：インプット／入力するという意味の英語

《使い方》行番号 INPUT 〝メッセージ*〟；変数名

変数には数値を入れる数値変数と，文字や記号を入れる文字変数の２種類がある。

A＝10 のようなものを代入文というんだけれど，この時のAが変数（この場合は数値変数）で，10という数字を代表することになる。

変数には，アルファベット26文字を単独，あるいは組み合わせて使える，１文字めをアルファベットにして２行め以降を数字にすることもできる。

ただ，はじめの２文字しか区別しない機種もある。その場合には，ＡＢＣとＡＢＤなんていう変数は同じものとして扱われてしまう。ベーシックの命令語をスペルの中に含む変数を使えないマイコンもある。注意しよう。

文字を代入する変数は文字変数といい，Ｘ＄のように，

＊ INPUTの後の〝 〟で囲まれた部分は省略することもできる。
このような省略できる部分も命令にはあるんだ。



$（ドルマーク〈ダラーと読む〉）をつけ，入力する文字は〝（ダブルクォート・引用符〈ふつうチョンチョンと言うことが多い〉）で囲む。これは約束だ。

例　X＄＝〝ランダム，エルシー　&　ラン〟

数値変数に文字を代入しようとすると，エラーになる。

10　INPUT　X＄

と入力して　RUNさせると，ディスプレイには「？」とだけ表示される。X＄に何か文字を入れてやるのを待っているんだ。だけど，これだけじゃ，プログラムを組んだ人以外，何を要求されているのかわからない。そこで，INPUT文にメッセージを書いておくと便利だ。

10　INPUT〝X＄＝〟；X＄*

と入力してRUNさせてみよう。

X＄＝？　と表示されたね。

そこで，RUN→RETURNで「シレイコード　ヲ　イレナサイ」のようなかたちの入力要求をさせるためには，

10　INPUT〝シレイコード　ヲ　イレナサイ〟，X＄*

と書けばよいワケだ。

文字変数に文字や記号を代入するときは，文字や記号をダブルクォートで囲まなくてはならないけれど，INPUT文で要求された入力の時には，その必要はないんだ。乱もただ「カワイコチャン」とだけ入力している。

だけど，乱の入力した文字は，はたして，マイコンの要求する指令コードなんだろうか……？

* INPUT文で〝　〟とX＄の区切りに；（セミコロン）を使うと？も表示されるが，，（カンマ）を使うと？は表示されなくなる。



第2種マイコン遭遇

PART III
IF～THEN～(その1)
PRINT & GO TO

「指令コードが違います!!」とコンピュータからのメッセージをカワイコチャンが伝えてきた。案の定，乱の入れた文字は違っていたようだ。カワイコチャンは再び「指令コードを入れなさい」といってきた。さて，これをマイコンで同じようにやるために，やるべきことを整理しよう。

①* 指令コードを入れさせる
② 入れた指令コードが正しいかどうか調べる
③ 違っていたらもう一度①にもどる
④ 合っていたら次の行動に移る

このようになればいいわけだ。

このうち，①は前のページでINPUTという命令を使うと便利だということがわかった。問題は②，③，④だ。これを解決するために便利な命令を一つ覚えよう。

IF～THEN～：イフ～ゼン～／もし～なら～せよ

《使い方》IF　条件　THEN　やりたいこと

ふつう，上のようにやりたいことを表したプログラムを文ということもついでに知っておこう。このIF文は，条件としてあげたものを満たしていれば*THENの後に指示しているやりたいことが実行される。もし，条件が満たされないと，THENの後のやりたいことは実行せずに，次のプログラムへ移っていく。さて，この条件だが，いろんなものがあるけれど，まず，次のものを覚えておこう。

＊　ここで①，②，③，④と順序よくやるべきことを箇条書きにすると便利だね。この本格的なものをフローチャートという。

＊　条件を満たすことをふつう真とよび，満たさないことを偽という。この言葉は数学で出てくるね。



《条件*の実際の書き方》

IF　A＝B　THEN　やりたいこと

というふうに使う。この場合AとBが同じ値であれば正しいから，やりたいことが実行されるわけだ。

今，乱が取り組んでいるのは，正しい言葉を入れることだから，この場合は，

IF　X＄＝〝求めているコード〃THEN～

となるわけで，このX＄がINPUTのところで乱が入れた言葉にあたる。

上のカワイコチャンは，乱が正しくないコードを入れたものだから，THENの後の～をやらずに，次のプログラムで「指令コードが違います」と言ってきた。つまり，

10* INPUT〝シレイコード　ヲ　イレナサイ〃;X＄

20　IF　X＄＝〝求めているコード〃THEN～

30　「指令コードが違います」と表示

40　行番号10番にもどってもう一度やり直し

となるわけだ。この30行でやるような表示を出すには，PRINTという大切な命令を使う。

PRINT：プリント／写すという意味の英語だね

《使い方》PRINT〝シレイコードガチガイマス〃

これで30行はOK。40行でやるように10行にもどるには，

GO　TO（GOTOでもいい）：ゴートゥ／～へ行く！

《使い方》40　GO　TO　10

だんだんプログラムができてきたぞ！

＊　条件にはこの他，A＜B，A＜＝B，A＞B，A＞＝B，A＜＞B（A≠Bのこと）等がある。数学と似てるね。

＊　このように，プログラムを作るときは，わかっていることから順に整理していくといい。



第2種マイコン遭遇 PART Ⅳ END

デタラメにコードを入れても絶望的だ。エルシーの言うように，プログラムのリストを出させて，コード名をチェックするというテもある。

さて，そのリストをちょっと調べてみよう。

```
10 INPUT "ｼﾚｲｺｰﾄﾞ ｦ ｲﾚﾅｻｲ",X$
20 IF X$="ADA" THEN 50
30 PRINT "ｼﾚｲｺｰﾄﾞｶﾞﾁｶﾞｲﾏｽ"
40 GOTO 10
50 PRINT "ADAﾗﾝﾄﾞﾍ ﾖｳｺｿ!! ﾄﾞｳｿﾞ
   ｵﾊｲﾘ ｸﾀﾞｻｲ"
60 END
```

もう，10行～40行の意味はわかるね。指令コードはどこにあるだろう？　20行だ。さっき覚えたＩＦ文の中だ。

Ｘ＄は10行で乱が入れた文字をたくわえる文字変数。この場合は，〝カワイコチャン〟が入っているね。それを20行の〝ADA〟と比べているのだから，違うのも当然だ。これで，指令コードはADAということはわかったのだけど，乱の入れたadaではだめだったようだね。その通り！　マイコンは〝ADA〟と入れることを求めてきたら，全く同じものでないとダメなんだ。乱の言うように，少しは大目にみて欲しいところだけど，そこはガマンして，このガンコ者とつきあっていこう。

さて，ちゃんと〝ADA〟と入れたらどうなるんだろう。IF文の約束では，THENの次に書いたものが実行されるね。



この場合は，50とだけ出ているね。これは，

IF X＄＝"ADA" THEN GO TO 50

と書いたのと同じで，行番号50へ行けということなんだ。

50行のPRINT文で何を出そうとしているかは，リストを見ての通りだ。そしてプログラムの終わりはEND。

END：エンド／終わりだ！ まさにジ・エンド

《使い方》行番号 END

行番号はプログラムの一番下になるのがふつう。そうでない場合もあるが，それは後のお楽しみ。

ところでADAって変な言葉だね。これは「エイダ」って読むんだ。実はこの言葉ちょっとした意味があるんだよ。

その一つは，英国のチャールズ・バベッジが作った世界最初の計算機械にプログラミングした，世界最初のプログラマーの名前なんだ。しかも，このエイダは，有名な詩人バイロンの娘なんだよ。

そして，もう一つは，米国の国防省が開発したプログラミング言語の名前であるということ。もちろん，上のエイダにちなんで付けられたのだけど…。このエイダという言語は，軍事関係の目的から，船や航空機，通信機器などに組み込んで使う計算システムのために作られたものなんだ。しかし，最近では，この他にいろんなものにも応用できる言語としてさらに成長しつつあるんだよ。

そして，最後のエイダの意味は？ どうやらこの物語の重要なカギを握っているようだ。



第2種マイコン遭遇 PART V IF～ THEN～（その2）

これでキミも一本の小さなプログラムを理解することができた。それは，

```
10 INPUT "シレイコート゛ ヲ イレナサイ",X$
20 IF X$="ADA" THEN 50
30 PRINT "シレイコート゛カ゛チカ゛イマス"
40 GOTO 10
50 PRINT "ADAラント゛ヘ ヨウコソ!! ト゛ウソ゛
   オハイリ クタ゛サイ"
60 END
```

だったね。これをRUNさせれば，「シレイコード ヲ イレナサイ」という表示が出て，もしADAと打ち込めば「ADAランド ヘ ヨウコソ!!……」のメッセージが出てくるというわけだ。乱たちは，まさにこのプログラムに遭遇したのだ。

ところで，プログラムの作り方はもうこれで十分だろうか？　もともと，プログラムは文章と同じで，同じことをさせるのに，いろんな表現の仕方がある。たとえば，キミたちが大人になって，プロポーズする時……。

あなたは私の太陽だ！

あなたはバラのように美しい……。

………

同じお墓に入りませんか？ etc.

と，彼女のハートにジーンと訴えるやり方はいくらでもあるハズだ。ただし，上の言葉で彼女にプロポーズできるか



どうかは，マイコンボーイは保証しない。

プログラムだって同じこと。別な命令を使ったり，同じ命令でも，使う順序などを変えることで，同じことをする別のプログラムができるんだ。左のプログラムを例にとって，別な書き方を試しにやってみよう。

このためには，新しい言葉を一つ覚えよう。ELSE*だ。

ELSE：エルス／～以外っていう英語の意味

このELSEは，これだけでは使わない。IF文と一緒に使うことになっていて，

《使い方》IF Ⓐ THEN Ⓑ ELSE Ⓒ：イフ ゼン エルス／もしⒶならⒷをやる，それ以外はⒸをやる

という使い方をするんだ。さて，これを使えば，

──もし，X＄が〝ADA″なら〝ADAランド へ……″を表示。それ以外は〝シレイコードが違います″と表示──

という形にプログラムを作ることもできるわけだ。

```
10 INPUT "ｼﾚｲｺｰﾄﾞ ｦ ｲﾚﾅｻｲ    ",X$
20 IF X$="ADA" THEN  PRINT "ADAﾗﾝﾄﾞ
   ﾍ ﾖｳｺｿ!! ﾄﾞｳｿﾞ ｵﾊｲﾘ ｸﾀﾞｻｲ"
   ELSE PRINT "ｼﾚｲｺｰﾄﾞｶﾞ ﾁｶﾞｲﾏｽ"
   :GOTO 10
30 END
```

ところで，ELSEの後ろに：GOTO 10ってあるけど，この**：(コロン)は1行に命令をまとめて書ける**役割を持っていて，上ではX＄が〝ADA″でない場合，PRINT命令とGOTO命令の2つを実行することになる。

＊ ELSEはエルスって読むけれど，人になぞらえてエルシーって読めば……。彼女の秘密が少しわかったかな？



第3種マイコン遭遇 PART I 配列（その1）

コンピュータはもともと，いろんな計算を何千回何万回とやる*ことが得意だ。マイコンでもそうで，人間だったらすぐにへたばってしまうようなこともヘッチャラだ。

とはいえ，そのような計算ももとはといえば人間サマが作ったプログラムを忠実に実行しているだけの話。計算の手順などをいちいち教えてやらないといけない。ところで，この計算を教えてやる人間の身になってみよう。いろんな変数があるし，データも山ほどあるし……，なんていうのはザラだ。

さて，そんな時どうしよう？

そういう時，いつでも，まず整理*できないか？　ということに立ち帰ってみよう。これはプログラムを作るうえでとても重要なことだ。

山ほどの変数やデータ。これを整理する方法があれば楽にならないか？　ということで考え出されたテクニックに**配列（英語でアレイという）**がある。これは，同じような仲間に番号を付けて区別する方法とでも思えばいい。

学校のゲタ箱に番号がふってあるね。たとえば，16番は僕のだなんていうふうに覚えているわけだが，このやり方と同じだ。ただ，書き方にルールがあって，

A＄(15)＝〝タロウ〟：A＄(16)＝〝マイコンボーイ〟：A＄(17)＝〝ハナコ〟というふうに文字の場合は書く。これで

＊　カッコヨクいえば，コンピュータがいろんな仕事をすることを処理するというんだ。

＊　プログラムには，考えの整理，使う命令の整理，変数の整理…と，いつも整理してかかることが大切だよ。



A＄の16番といえばマイコンボーイのことだとわかるようにするわけだ。数字の場合は，変数のとき話したのと同じように，＄をとればいい。

A(15)＝25：A(16)＝99：A(17)＝77

などと書き，Aの16といえば99という数字を示すことになる。さらにこの配列には2次元というものもある。これは

	タロウ	マイコンボーイ	ハナコ
国　語	75点	－10点	85点
英　語	82点	－100点	68点

などというものを整理する時に使う。A(科目，名前)＝……，あるいは，A(名前，科目)＝……，といったふうに書く。たとえば2番目のマイコンボーイの1番目の科目の国語は，

A(2，1)＝－10となる！

このような配列はプログラムの中でよく出てくるので，少しずつ慣れるようにしよう。

また，配列を使う時は，マイコンに，配列を使いますよという宣言をしてやる必要があるんだ。この時使うのが，

DIM*：ディメンション／dimentionの略で次元の意味

《使い方》DIM　A(10)，B(1，3)，X＄(5，10)

A，B，X＄は配列を代表する名前で，カンマで区切れば，一度にいくつも宣言できる。カッコの中の数はその配列で必要な数を入れるのがふつうだ。

＊　マイコンの命令にはこのように単語や語句の省略形も多い。こんなときは元の単語などで覚えるとわかり易いぞ。



第3種マイコン遭遇 PART II FOR～NEXT & READ～DATA

何やら変なことをしゃべるマイコンロボットが出てきた。このロボットのリストをみると…,

```
10 DIM A$(4,10)
20 FOR I=1 TO 10
30  FOR J=1 TO 4
40   READ A$(J,I)
50  NEXT J
60 NEXT I
120 DATA ｺ,ﾉ,ﾙ,R,ﾝ,ﾁ, ,S,ﾋﾟ,ｭ,ｹﾞ,
    2,ｭ,ｳ,ｰ,3,ｰ,ｽ,ﾄ,2,ﾀ,ｳ,ﾉ,C,･,ﾆ,
    ﾅ,ﾃﾞ,ｴ,ﾂ,ﾏ,ｱ,ｲ,ﾅ,ｴ,ﾙ,ﾀﾞ,ｶﾞ,ﾊ,!
```

また見慣れない命令が出てきたようだ。

まず最初はFOR～NEXT文だ。これは,

FOR～NEXT：フォー　ネクスト／～で決めた範囲でのくり返しを行う

《使い方》行番号　FOR　変数＝初めの値　TO　終わりの値

行番号 ⋮ } くり返しさせたいことを命令する

行番号　NEXT　変数

これは，変数の値が初めの値から終わりの値になるまで1ずつ増えていきながらFORとNEXTの間をくり返し行う役目を持っている。

マイコンロボットのリストを見ると，FOR　I＝1……というのと，FOR　J＝1……という2つのループ*が出て

＊　バグ＝Bugと英語では書く。もともと虫のことで，プログラムの世界ではプログラム中のエラーのことを言う。

＊FOR NEXTのようにその間を何度もくり返すようなプログラムはループと呼ぶ。



いるね。しかも　FOR　I＝………NEXT　I　の内側にすっぽり入っている。こういうのを「入れ子」ということを覚えておこう。よく使うテクニックだ。これで上のプログラムを見ると，Iが１増えるたびに，Jが４回ぐるぐる回ることになる。そしてIは，10回ぐるぐる回るからREAD　A＄(J，I)という仕事は４×10＝40回もやっていることになるね。

ところで，このREADって何だろう。A＄(J，I)はもう出てきたから覚えてるかな？　文字の２次元配列だ。

READ*：リード／読むっていう英語

DATA　：データ／もうこれは日本語になってるね

この２つは，FOR～NEXTと同じようにペアで使う。

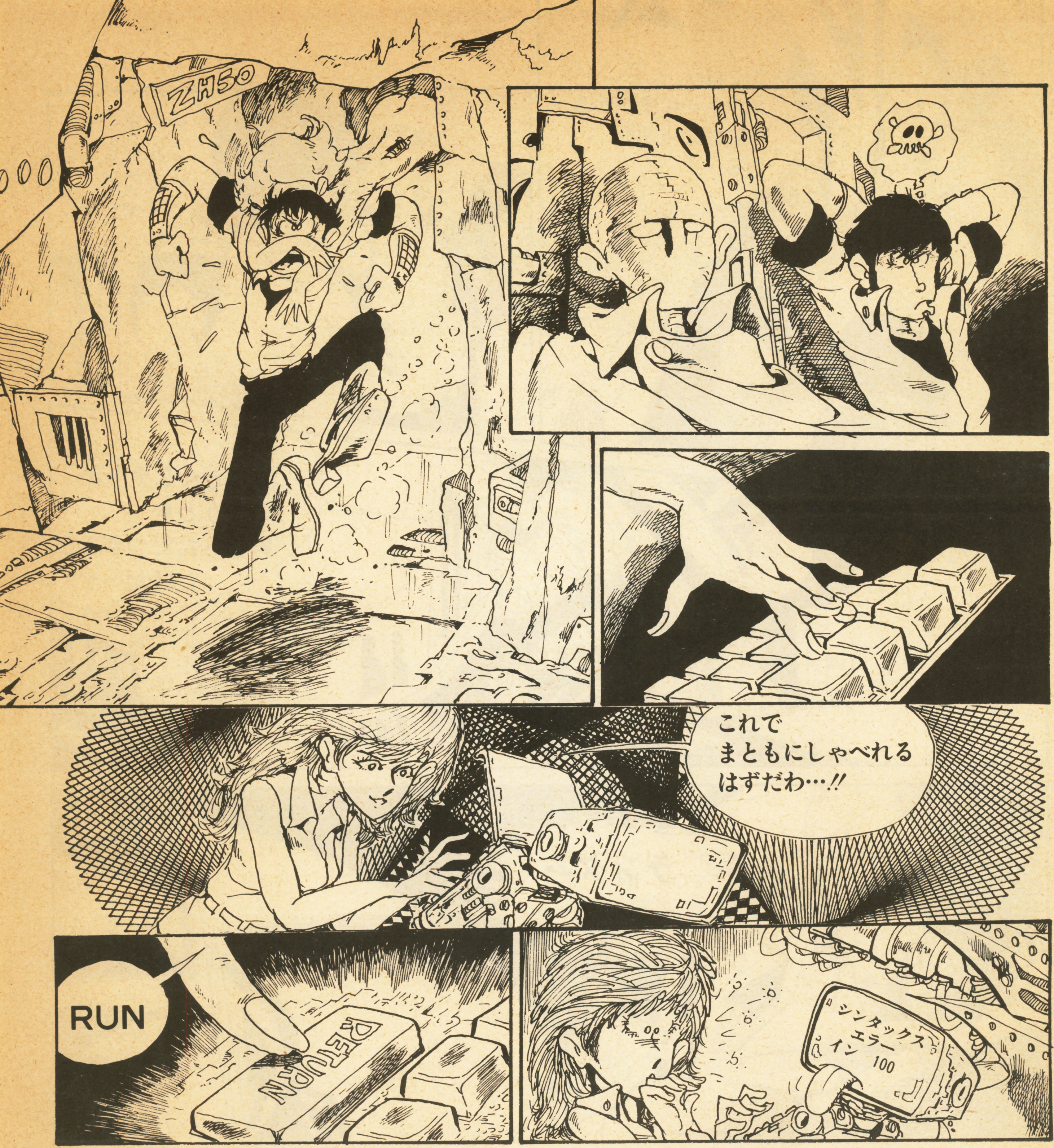
《使い方》行番号　READ　変数１，変数２，……

行番号　DATA　データ１，データ２，……

１回READ文が実行されると，後ろに指定した変数にそれぞれ，DATA文中のデータが読み込まれる。変数１にはデータ１が，変数２にはデータ２が，……というぐあいだ。だから，読み込む変数とデータの数が同じであることがまず基本となるね。

このDATAは数字であっても文字であってもかまわない。どうやら上のマイコンロボットは，FOR～NEXT文と２次元配列を使ったREAD～DATA文で何か秘密をたくみにかくしているようだ。カギはDATAの並べ方にありそうだけれど……。

＊　READ～DATAのような命令をみると，プログラムって易しい日常英語がもとになっていることがわかる。



第3種マイコン遭遇 PART III スクリーン・エディット

マイコンにプログラムを実行させようとした時，もしプログラムに間違いがあると，ディスプレイにエラーメッセージ*が表示される。エラーメッセージにはいろいろあるが一番代表的なのが，マイコンロボットの表示した，

Syntax Error in 行番号（シンタックス　エラー　イン　行番号　と読み，指定した行番号に文法ミスなどがあるということを示す）だ。また，

Next Without For in 行番号（ネクスト　ウィズアウト　フォー　イン　行番号　と読み NEXT に対応する FOR文がないという意味）。これは，エルシーが FOR～NEXT文の入れ子を正しく書かずに，FOR文どうしがクロスするようにしてしまったためだ。

さて，エラーメッセージが表示されたらどうすればいいだろうか？　プログラムを修正するということが必要になるね。このようなとき，マイコンにはスクリーン・エディット*と呼ばれる便利な機能があるんだ。

やり方は，LIST命令でプログラムのリストを画面に表示させてから，その画面上で文字等を直すわけだ。このようなことをするために，いくつかのエディットキーと呼ばれるキーがマイコンにはついている。[DEL]，[INS]，[←]，[→]，[↑]，[↓]　などと書いてあるキーのことだ。これらのキーの使い方を具体的に紹介しよう。

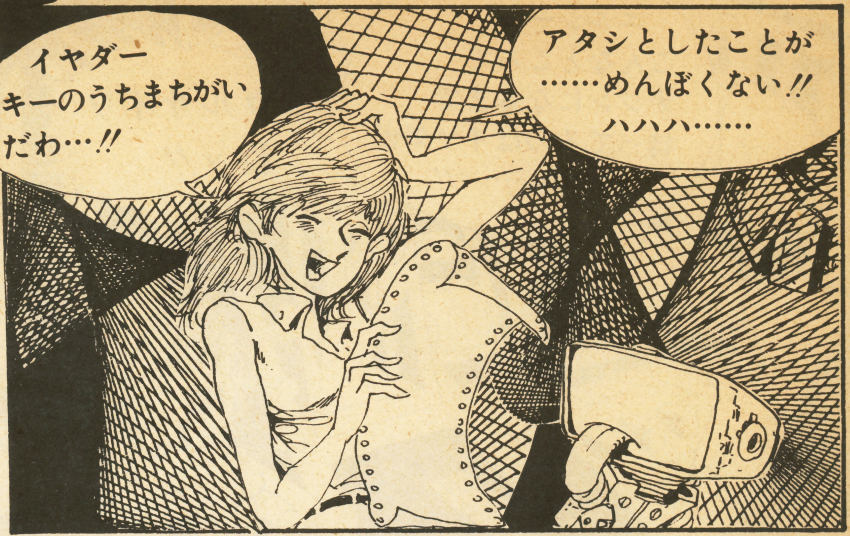
＊　エラーメッセージとは，間違ってますよ！っていう内容を伝える意味だ。

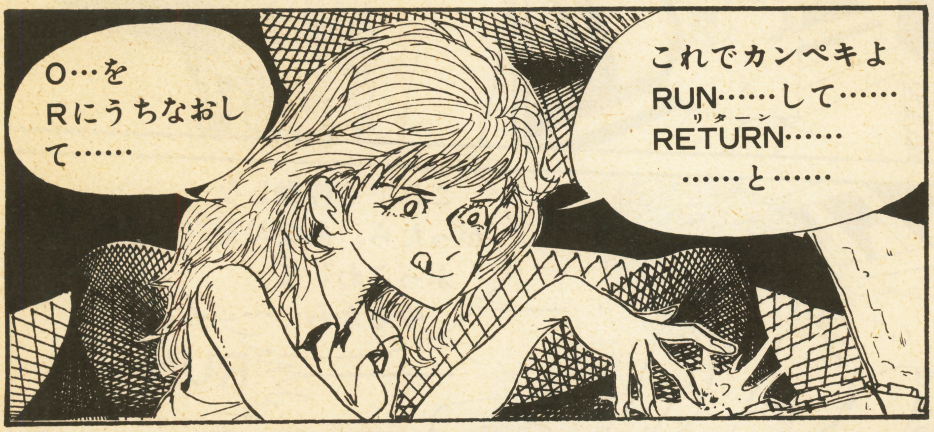
＊　スクリーンは画面。エディットは編集。合わせて画面編集という。つまり，テレビ画面上でプログラムの直しをやるわけだ。



LIST

```
70  FOO J=1 TO 4
80   FOR I=1 TO 10
90    PRINT A$(J,I);
100  NEXT J
110 NEXT I
120 DATA コ,ノ,ル,R,……
```







このように
FORとNEXTの対は
交差するとエラーになる!

```
┌→70  FOR J=1 TO 4
│┌→80   FOR I=1 TO 10
││ 90    PRINT A$(J,I);
└┼→100  NEXT J
 └→110 NEXT I
120 DATA コ,ノ,ル,
    R,ン,チ, ,S,
    ピ,ュ,ゲ,2,ュ,
    ウ,ー,3,ー,ス,
    ト,2,タ,ウ,ノ,
    C,・,ニ,ナ,デ,
    エ,ツ,マ,ア,イ,
    ナ,エ,ル,ダ,ガ,
    ハ,!
```

《カーソル移動キー》 ←, →, ↑, ↓

マイコンのスイッチをオンすると，ディスプレイの左上に点滅するものがある。これはカーソルと呼ばれるものだ。キーを適当にたたいてみよう。次の文字が書かれる場所にいつも点滅しているね。このカーソルを移動キーで直したい文字の上に動かし，新しい文字を打ち込めば，その部分が書きかえられるというわけだ。ただし，直した後，必ずリターンキーを押すことをお忘れなく。

《デリートキー》* DEL／delete（デリート）削除するという意味の英語の省略形

よけいな文字を削除して右側の文字をつめたいときに使う。カーソルを消したい文字の上に持っていって使う。

《インサートキー》 INS／insert（インサート）挿入するという意味の英語の省略形

デリートキーの逆でカーソルのある文字の右側がキーを押すたびに一つずつあいていく*。割り込ませたい文字を入れるにはこれを使うと便利だね。

デリートキーかインサートキーを使ってプログラムを直した後は，必ずリターンキーを使わないとマイコンには伝わらない。注意しよう。

マイコンロボットのプログラムにエルシーが書き加えた150行から190行までには2つのエラーがあったようだ。

150行のFORがFOOに，もう一つは2重のFOR文が交差して入れ子になっていないことだね。エディット開始!

*機種によっては少し異なることがある。

* 機種によっては，もう一度押すまでずっと挿入されるものもある。このような時インサートモードにあるという。



第3種マイコン遭遇 PART Ⅳ 配列（その2）

マイコンロボットが持っていたプログラムリストには，A＄（J，I）という配列があったね。これは，前にチョットだけふれた2次元配列だ。

《A＄（J，I）の配列表》

J＼I	1	2	3	4	5	6	7	8	9	10	…
1	コ	ン	ピ	ュ	ー	タ	・	エ	イ	ダ	
2	ノ	チ	ュ	ウ	ス	ウ	ニ	ッ	ナ	ガ	
3	ル	…	…	…							
4											
⋮											

A(1, 1)＝コ
A(1, 2)＝ン
A(2, 1)＝ノ
A(2, 2)＝チ

左のようなしくみで，データをしまっていくんだ。数学で習う座標と同じに考えればよい。

マイコンロボットがはじめに持っていたプログラムを，もう一度見てみよう。

```
10 DIM A$(4,10)
20 FOR I=1 TO 10
30  FOR J=1 TO 4
40   READ A$(J,I)
50  NEXT J
60 NEXT I
120 DATA コ,ノ,ル,R,ン,チ, ,S,ピ,ュ,ゲ,
    2,ュ,ウ,ー,3,ー,ス,ト,2,タ,ウ,ノ,C,・,ニ,
    ナ,デ,エ,ツ,マ,ア,イ,ナ,エ,ル,ダ,ガ,ハ,!
```

二重になっているFOR～NEXTループ の外側のループ



をＩのループ，内側のループをＪのループと名付けよう。

変数Ｉの値は，Ｊのループが４回まわるあいだ，つまりＪが初期値１から終値４まで変化するあいだＩ＝１だ。

40 行のREAD文は，Ｉのループが１回まわるあいだに，Ａ＄(１，１)，Ａ＄(２，１)，Ａ＄(３，１)，Ａ＄(４，１)の各配列要素に，120 行のデータを前から順番に読み込んでいくというしくみだ。

そして，Ｉのループが２回めに入れば，Ａ＄(1，2)，Ａ＄(2，2)，……と続み込んでいくんだ。

というふうにして，左ページの表のように文字がしまわれる。たてにたてに読み込んでいく，というしくみだ。

これを横方向に書き出していくプログラムをエルシーが考えたというわけだ。

エルシーの入力したプログラムを実行すると，

Ａ＄(１，１)，Ａ＄(１，２)，Ａ＄(１，３)，……の順番で表示されていく。

二重のFOR～NEXT 文の内側と外側の変数を入れかえることで，暗号を解読できたんだ。

まあ，エルシーの作戦も，あてずっぽうに近いやり方だけど，いちばんオーソドックスな解読法だからね。

「コンピュータ・エイダノ　チュウスウニツナガル ゲートノナマエハ RS232C デアル！」

ドアの番号は，RS232C* というわけだ。ところで，あとをつけてくるのはいったい誰なんだろう……。

＊RS232C というのはコンピュータどうしが通信をしたりする時よく使われるんだよ。



第3種マイコン遭遇 PART V 計算せよ!

なんだかガメツそうな男がでてきたね。カネ勘定が得意そうだ。ところで，マイコンはふつうの電卓と同じように計算機能も持っているんだ。

マイコンと電卓に，3×(2－1)＋6÷2の計算をさせてみよう。どう違うかな。

電卓 [2][－][1][＝]→ 表示1……1をメモ

[3][×][1][＝]→ 表示3……3をメモ

[6][÷][2][＝]→ 表示3……3をメモ

[3][＋][3][＝]→ 表示6 ➡ 答え 6

こまかく区切ってメモをとる必要がある。

マイコンでは，×(カケル) のかわりに＊(アステリスク) ÷(ワル) のかわりに／(スラッシュ) を使う。

マイコン (ダイレクトモードでやってみよう)

[P][R][I][N][T][3][＊][(][2][－][1][)][＋][6][／][2]

→ [RETURN]

ディスプレイに上のように書いて [RETURN] キーを押してやれば，計算の途中でメモをとらなくても，答えが1回ででてくる。

この違いは，マイコンの計算記号*(＋，－，＊，／，∧，¥，MOD)に，実行する場合の優先順位があるからなんだ。右ページに表にまとめてみた。もちろん，カッコは，すべての計算記号に優先する。数学と同じだね。

＊計算記号はムズカシク言えば算術演算子と呼ばれている。



優先順位	算術演算子	説明
1	∧	べき乗：3∧2 は 3^2 と同じ
2	*	かけ算
	／	わり算
3	¥	特殊なわり算 小数点以下を切り捨てて答えを表示する
4	MOD	わり算の余りを整数で表示
5	＋	たし算
	−	ひき算

¥＝通称「エン」マーク　MOD＝通称「モッド」

こんなふうに数学と同じ優先順位が決められているので紙に式を書くようにディスプレイに書いてやれば，マイコンが自分で計算してくれるんだ。

¥とMODは，ふつうの式では使わない演算子だ。例をみてみよう。

〈例1〉 P R I N T　1 0 ¥ 3　RETURN

答え　3

3.3333……の小数点以下が切り捨てられたワケだ。

〈例2〉 P R I N T　1 1 . 2　M O D 6

→ RETURN

答え　5

わり算の余りは，小数点以下が切り捨てて表示される。



第3種マイコン遭遇 PART VI プログラム・ストップ

乱は，いったいどんなふうにプログラムを書きかえたのだろうか？ リストを見てみよう。

```
10 INPUT "シレイコート゛ヲイレナサイ";X$
20 IF X$="ランハエライ" THEN 50
30 PRINT "ツラヲアラッテテ゛ナオシテコイ"
40 GOTO 10
50 PRINT "ランハイイオトコ"
60 GOTO 50
```

まず，20行の指令コードが〝ADA〟から〝ラン ハ エライ〟に変わっている。そして，その指令コードを入れると50行へ行く。そして〝ラン ハ イイオトコ〟と表示して60行へ。60行へ進むとまた50行にもどる……，といつまでたっても終わらない。

GOTO文は気をつけて使わないと，このような無限ループに陥ってしまうことがある。プログラムを作るときはこのようにグルグル回りっぱなしの部分がないかどうかを十分注意しよう。

もっとも，上の場合は，わざと乱がやったんだけどね。

ところで，このように無限ループに陥って，マイコンが止まらなくなったらどうしよう？ 心配しなくてもだいじょうぶ。ちゃんとそのような無限に続く命令を中止させるキーが用意されているんだ。

STOP：ストップ／止めるという意味だね



BREAK：ブレイク／中止するという意味がある

CAN：キャンセル／取り消すという意味のcancelの省略形

のいずれかのキーがあるはずだ。このような時は，これらのキーを押そう。すると，

Break in 行番号：ブレイク　イン　行番号

と，キーを押した時に実行中だった行の番号を表示して，プログラムは実行を中止する。

さて，これで乱はRS232Cを通って新たなアドベンチャーへ向かうことができた。この間，乱と共に覚えた命令をもう一度おさらいしてみよう。

NEW：ニュウ／プログラムを消去する

LIST：リスト／プログラムの一覧を出す

RUN：ラン／プログラムを実行する

INPUT：インプット／変数に文字や数字を入れる

IF Ⓐ THEN Ⓑ ELSE Ⓒ：イフ Ⓐ ゼン Ⓑ エルス Ⓒ／もしⒶならばⒷを実行。それ以外はⒸを実行。

PRINT：プリント／ディスプレイに表示する

GOTO：ゴウツー／行番号 ～ へ行く

END：エンド／プログラムの実行終わり

DIM：ディメンション／配列の使う大きさを確保

FOR～NEXT：フォー　ネクスト／くり返し実行

READ ～ DATA：リード　データ／データを変数に読み込む

BREAK IN 36

プログラムの基本ルール

マイコンに仕事をさせるには，仕事の内容をこまかくきざんで命令する必要があったね。そしてこまかくきざんだ命令を実行する順序の目印に行番号を付けることにしていたね。行番号と命令とで，マイコンは，自分が何をすればいいのか理解するわけだ。

プログラミングとは

実際にプログラムを組む時の手順を考えてみよう。

キミたちは，今までにいくつかの命令を学んできた。これからもいくつか命令を学んでいくけれど，プログラミングする時に必要なのは，たくさんの命令を知っていることじゃない。

まず第一に，自分がマイコンに何をさせたいかを整理すること。

第二に，整理したものが，自分の知っている範囲内の命令をどう使えばできるかを考えてみることだ。

実際今までに覚えた命令をベースにするだけでもチョッとした仕事ができる。ためしに，分と秒を表示するデジタル時計を作ってみよう。

デジタル時計のプログラム

まず最初に，デジタル時計の時間表示のしくみをこまかく整理しよう。

① 1秒ごとに秒の表示を1ずつ増やす。

② 秒の表示は0から59まで増えて次は0にもどる。

③ 秒の表示が0にもどるたびに，分の表示が1ずつ増える。

④ 表示の位置は，分の右側に秒。

⑤ 分と秒のあいだは：(コロン)で区切る。

さあ，上で5つに分けた仕事は，それぞれどんな命令を使えばできるかを考えてみよう。

①と②は，FOR～NEXT文とPRINT 文を使えばよさそうだ。変数の初期値を0，終値を59に設定すればいいね。ただし，1秒に1回ループをまわす必要があるから，時間つぶしにループの中で，別のループを何回かカラまわりさせてやる必要がある。

③はどうだろう。分の表示を1ずつ増やすためには，やはりFOR～NEXTが使えそうだ。

分の表示がかわるごとに，秒のループは同じ仕事をくり返すワケだから，分と秒のFOR～NEXT を2重ループにすればできそうだ。

⑤のコロンは，秒の表示と一緒にしてみよう。

プログラミングのテクニック

④はどうしようか。これは，今までの命令ではできないんだ。1ずつ増えていく数字は，同じ位置に表示しなくてはならないからね。LOCATE文 という便利な命令がある。

LOCATE ：ロケイト／場所を決めるという意味の英語

《役　目》 表示位置の指定

《使い方》 LOCATE x，y

xは左から何文字めか，yは上から何行めかを指定する。

▼デジタルウオッチのプログラム

```
10 REM ﾃﾞｼﾞﾀﾙ ｳｵｯﾁ
20 FOR I=0 TO 59
30  FOR J=0 TO 59
40   LOCATE 10,10
50   PRINT I;":";J
60   FOR U=1 TO 670
70   NEXT U
80  NEXT J
90 NEXT I
```

リストを読みやすく

自分で組んだプログラムなら，どの行で何をさせているのかなんて当然わかってる。ところが，他人の組んだプログラムというのは，意外に読みづらいものなんだ。

最近みたいにいろいろな人がマイコンを使うようになってくると，他の人が組んだプログラムを読む機会も多くなりがちだ。そこで，デジタルウオッチもチョッと工夫してみよう。

●REM：レム／注目するという英語
REMARKの省略形

《役　目》 リストに注釈をつける

《使い方》 行番号　REM　注釈
　　　　　行番号　'　注釈

他人の組んだプログラムは，どこが何をしているのかわかりづらい。REM文で，プログラムリストの中に注釈をつけておくと便利だ。

REMは'（アポストロフィー）で代用できる。リストの中の行の間をあけて見やすくするのに使うといい。

●“入れ子”の書き方

エルシーが解読した暗号文には，FOR～NEXTの2重ループが使われていたね。

どのFOR文がどのNEXTとループになっているかは，そのループの変数名を見ればわかるけど，内側のループを1字下げて書くと一目瞭然(りょうぜん)だ。デジタルウオッチプログラムも，この点を考えて書こう。

●変数表を作る

プログラムではたくさんの変数を使う。どの変数が何を表しているかを表にしながらプログラムを組んでいくと，プログラミングもし易いし，あとで他の人が読むときもわかり易い。

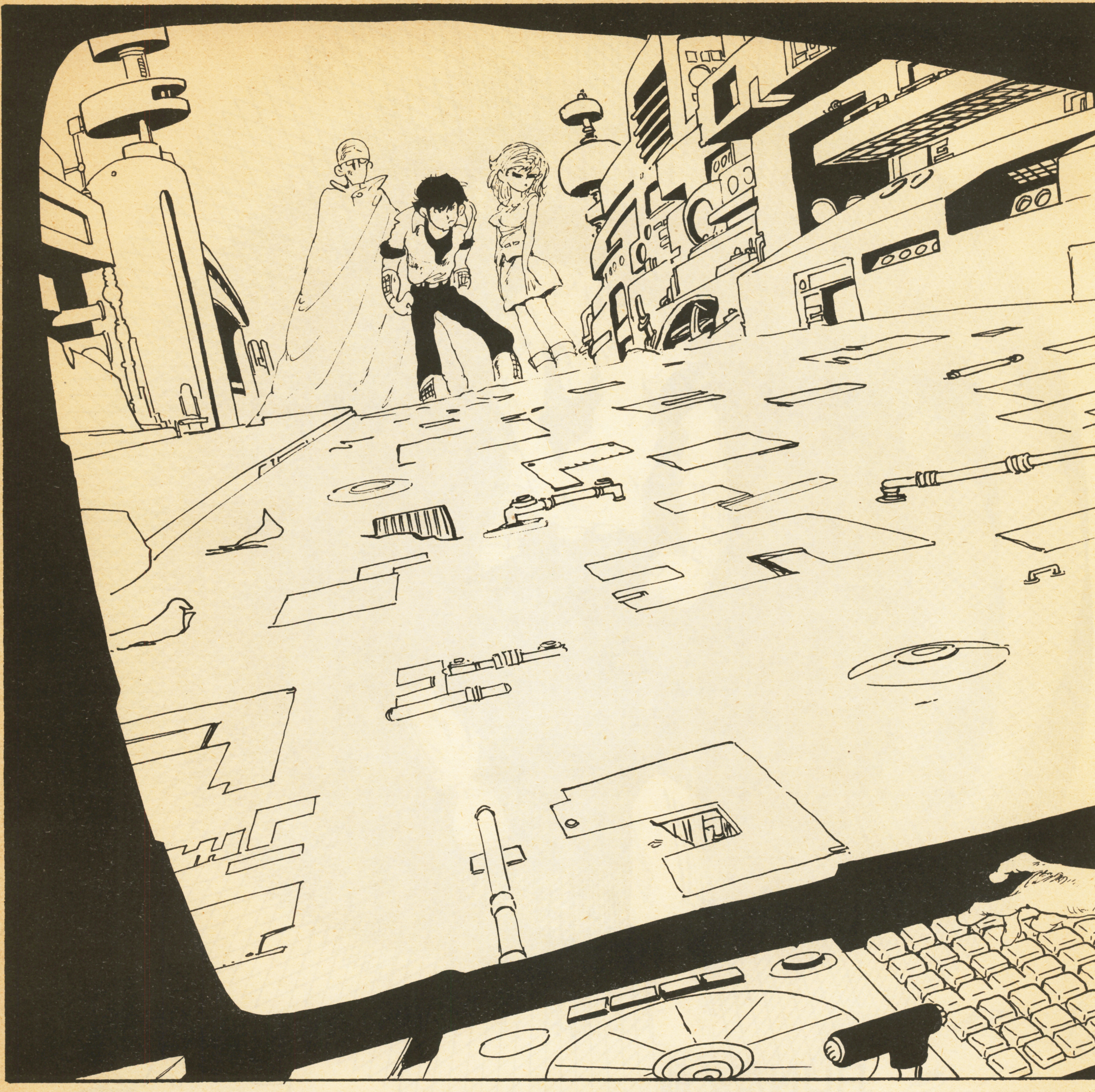
〈デジタルウオッチ変数表〉
例　I……分を表す。
　　J……秒を表す。
　　U……カラまわりループの変数を表す。

さあこれで，準備OK！ 実際にデジタルウオッチをプログラムすると左の通り，意外にシンプルにできただろう。



RUNだ!! 乱

第2章 電子地雷エレクトロマイン



ゲームプログラミング開始 キルとアルゴの挑戦

さて，いよいよ，マイコン遭遇で得たベーシックの知識をより本物にするためにゲームプログラミングに挑戦だ。

その第一弾は，人類を支配しようと凶悪化した超巨大コンピュータにあやつられるキルとアルゴ，この二人が乱たち一行にしかける電子地雷エレクトロマイン*。

それは，RS232Cの電子ベルト地帯に敷設した地雷原だ。この地雷原をくぐり抜けないことには，せっかく，エイダの中枢へ到る RS 232C ゲートを捜しあてた苦労も水の泡となる。

ところで，これからゲームを作っていくわけだが，まず，それに先立って，このエレクトロマイン・ゲームの概要を考えてみよう。作るべきゲームの内容をはっきりさせるわけだ。

ゲームは，乱，エルシー，ランダムこの３人が地雷をさけながら歩いて地雷原をこえるだけという単純なものにしよう。とはいえ，これだけではつまらないから，地雷はほんのわずかな時間しか見えないようにしてしまおう。つまり，一度地雷を見せてまた消すわけだ。こうなると，乱たち一行は記憶力だけが頼りとなる。また，前の人が歩いた跡は歩けないことにして，悪知恵の働きそうな乱たちがズルをできないようにしてしまおう。さらに追打ちをかけて制限タイムを設けよう。タイムオーバーになるとゲーム終

＊ マイン＝英語でmineと書き，地雷の意味。
I, my, me, mineのmineと同じスペル。



了となるわけだ。

ゲームの第1回目の地雷の数は4個。そして3人が全員地雷原をこえるごとに地雷の数を2個ずつ増やしていこう。ますます地雷原をこえることが難しくなるわけだ。このような難関をみごと3面終了したら突破成功と表示してゲームオーバーとしよう。そして実際に歩かせるのにはテンキーを使おう。さらに，ゲームとしてのおもしろさを増すために，毎回速く到着できれば，それだけ得点が上がるような工夫をした得点制もとり入れてしまおう。

以上が，ゲームの内容だ。だいたいどのように遊ぶゲームかイメージがわいたかな？

さて，これをもとに具体的にプログラミングするわけだが，マイコンボーイとしては，次のことを常に念頭におきながらゲームを作っていくことにしよう。

①ゲームの中で必要となるいくつかの表示，計算などの処理をこまかく区分けする。

②その区分けした一つ一つの処理を短い簡単なプログラムに作りあげる。

③そのようにしてできた一つ一つのプログラムを全体の中のどの位置に置けばいいかを考える。

この3点だ。

これは，どんなプログラムを作るときにも必要なことだから，アダ，オロソカにしないようにしよう。

では，プログラミング・スタート！



ゲームの構成を考える フローチャートを作る

アルゴがキルにエレクトロマインのプログラムの概要を示しているね。物語がいくつかの小さなストーリーからできているのと同じように，このゲームもいくつかの役目をもったプログラムの組み合わせでできている。この組み合わせをどのようにしたらプログラムがスムーズに進むのか調べるために，よく図式化することがある。アルゴは上でまさにそれをやっているわけだ。この図式化の本格的なものにフローチャート*というものがあるけれど，それを使って，ここで作ろうとしているゲームの組み立てをハッキリさせてみよう。

右ページに示したものがそのフローチャートだ。ふつうはフローチャートは上から下へ流れるのだけど，ここでは左から右へ進んでいくようにしてある。このフローチャートの中に使われている記号の意味は，

□ は中に書きこまれた仕事をする部分

◇ は中に書きこまれたことがあっているかいないかで枝分かれする部分。あっていればYesへ，あっていなければNoへ進む

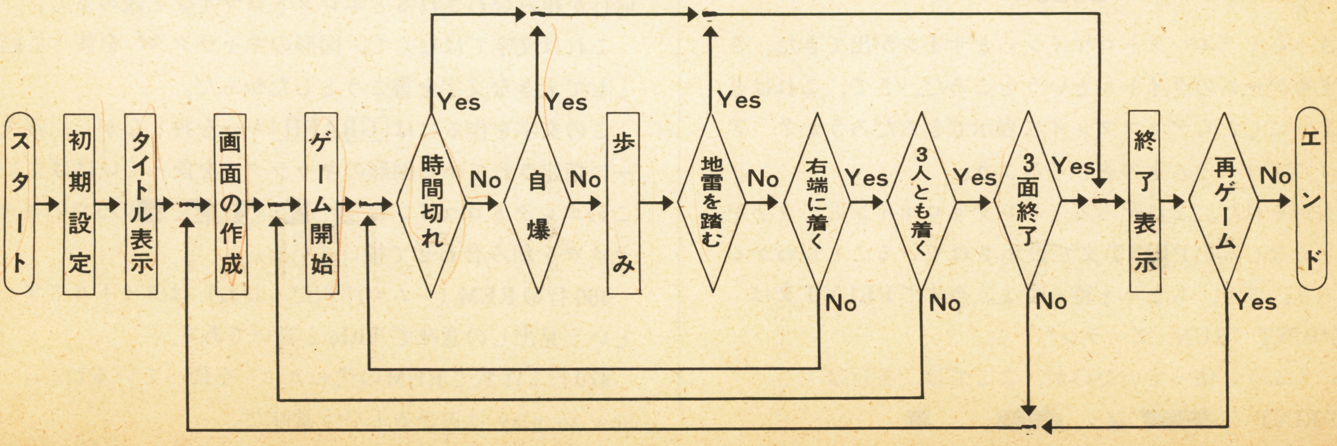
ということだ。

このフローチャートで，実際のプログラムの組み立てを十分検討することが，プログラミングの第一歩。どういうプログラムを組み合わせていけばいいかわかったかな？

＊フローチャート，よく耳にする言葉だね。流れ図という意味があるんだよ。



エレクトロマインのフローチャート（プログラムのおおまかな構成）





タイトルの表示 図形のキャラクタ COLOR

さっそく『エレクトロマイン』とサインが出てきた。さしずめゲームのタイトルというところだ。さて，これはどのようにしたらディスプレイに表示できるだろうか？ アルゴのプログラムをのぞいてみよう。

右がそれだ。これをみるとエレクトロマインという字が，もうおなじみのPRINT文で表示されていることがわかるだろう。でも，ちょっと待ってよ。今までPRINT文は

PRINT "RUN ダ ラン"

というふうに使っていたのが，ここでは，各行は

PRINT "▰◢▰ ■ ……"

のようなもので組み合わされている。でも，そんなものが何行か組み合わされるとエレクトロマインと読めるね。

これは文字ではなくて，図形のキャラクタ* を使って絵として大きな文字を書こうとしたからだ。

この文字を作るには GRAPH キーを押しながら，各キーに指定されている図形のキャラクタを使っていけばよい。このエレクトロマインは◢，◣，◥，◤，■の5つのキャラクタを組み合わせて描いている。

460行のREM（レム＝注釈だったね）はタイトルですよ，という見出しの意味でTitleと書いてある。

470行も注釈でREMの代わりに'を使っているね。how to play は遊び方という意味だ。

＊ キャラクタ＝character／文字，記号の意味がある。マイコンにはアルファベット，カタカナ以外にこういう図形も使える。



480行にはCOLORという命令が出ている。これはタイトルのエレクトロマインの色を指定するためのものだ。

COLOR：カラー／色という意味だね

《使い方》COLOR　表示したい色，背景の色*

ここで使える色は全部で8色あって，番号で指定する。この色と番号の関係は，

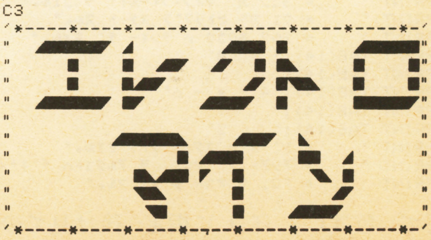
0……黒　1……青　2……赤　3……紫

4……緑　5……水色　6……黄　7……白

だ。よーく覚えておこう。アルゴのリストでは，色はC3と色番号を変数で指定しているね。これはたぶん表示する色を後で変えるのが楽なようにしたかったからだろう。というのも，C3＝3などとプログラムの最初に決めておいたとすると，その部分をC3＝1としてやるだけで，COLOR C3としたところはすべて青に変えられる。

さて，このプログラムが理解できれば，ゲームの終了時のENDの表示，突破成功の表示もプログラムできそうだ。全リストは74ページから出ているから見てみよう。

```
460 REM ---------- Title ----------
470 'how to play
480   COLOR C3
490
500   PRINT "
510   PRINT "
520   PRINT "
530   PRINT "
540   PRINT "
550   PRINT "
560   PRINT "
570   PRINT "
580   PRINT "
590   PRINT "
600
```



*　PC8001ではCOLOR文中，背景の色は指定できないので，COLOR 表示したい色――とする。



HOW TO PLAY PART I TAB

さて，タイトルの注釈文にhow to play* とあるからには，遊び方を示した部分があるはずだ。そのリストをのぞいてみよう。

```
610    COLOR C7
620    PRINT :PRINT
630    PRINT "     * 3ニン ガ マイン・フィールド ヲ コエル。"
640    PRINT "     * ウエ カラ ヒトリヅツ スタート スル。"
650    PRINT "     * アルク ホウコウ ハ テンキー デ キメル。"
660    PRINT TAB(14);"    8"
670    PRINT TAB(14);"    !"
680    PRINT TAB(14);"4 - * - 6"
690    PRINT TAB(14);"    !"
700    PRINT TAB(14);"    2"
710    PRINT "     * ジライ ハ スタート ノトキ ミエル。"
720    PRINT "     * アシアト ノ 「マウエ」 ヲ アルクコトハ デキナイ。"
730    PRINT "     * 「Q」 ヲ オスト ジメツ スル。"
```

610行で，色はＣ７（白）に指定。後は見ての通りだ。

いや，一つ見慣れないものがある。TABだ。

TAB：タブ／もともと一覧表を作るという意味がある

《使い方》PRINT　TAB（数）

このように，PRINT文中に一緒に使うようになっていて,* カッコの中の数だけ空白を出して，その後"で囲まれた文字を表示する役割を持っている。つまり，歩く方向を指示するキー（２，４，６，８の数字キー）の図の説明を14字分右にずらしているわけだ。どうやらこのゲームは歩く方向を２，４，６，８の４つのキーで指示するようだね。

次に３人を画面に出さなくてはならないわけだが，それについては，足跡の形で３人を表示することにしよう。画面に絵を出すのが少しは楽になるはずだ。

＊ how to＝～する方法，play＝遊ぶ。合わせて，遊び方っていうわけだ。

＊ TABはLPRINT（エルプリント＝プリンタに文字を印字する）文中に使うこともできる。



それから620行でPRINT*が2つコロンでつないで出ているね。要するにPRINT命令を2回実行しなさいということなんだが，表示する文字などを何も書いてない。こういう場合のPRINTは単に改行するだけ，つまり，

PRINT "ABC" →ＡＢＣを表示
PRINT →改行をする

ということなんだ。だから620行のPRINTは改行が2回，つまり2行改行して，それから"＊ 3ニンガ……"という文章を表示するというわけなんだ。

さて，これで42ページのタイトルの表示と合わせて，遊び方の説明をするというまとまったプログラムが1つできた。エレクトロマインゲームのタイトルと遊び方はこれを使えばOKというわけだ。



▲タイトルと遊び方を表示。

＊ PRINT文も使いようで，表示もすれば，改行もする。プログラムの命令はみーんな使い方しだいだ！



HOW TO PLAY PART II LINE, PAINTなど

さて，タイトル表示，遊び方の説明はできたけど，前ページの写真をみると左下に足跡が出ているね。この部分を作り忘れてた！ さっそく作ってみよう。

とはいえ，この部分はちょっとしたテクニックがいるので〝作る〞前に，新しい命令をまとめて勉強しておこう。

(810行) LINE：ライン／線という意味だね

《使い方》行番号 LINE（スタートのx座標，y座標）－（終点のx座標，y座標），線のひき方，線の色

上のx座標，y座標は，各マイコンごとに決まっている範囲内の数字で，FM8などではx座標が0～639まで，y座標が0～199まで使える。線のひき方はふつうの線の場合PSET(Point SET：ポイントセット／点を表示する)を使う。

(850行) PAINT*：ペイント／ぬるという意味

《使い方》PAINT（x座標，y座標），ぬる色

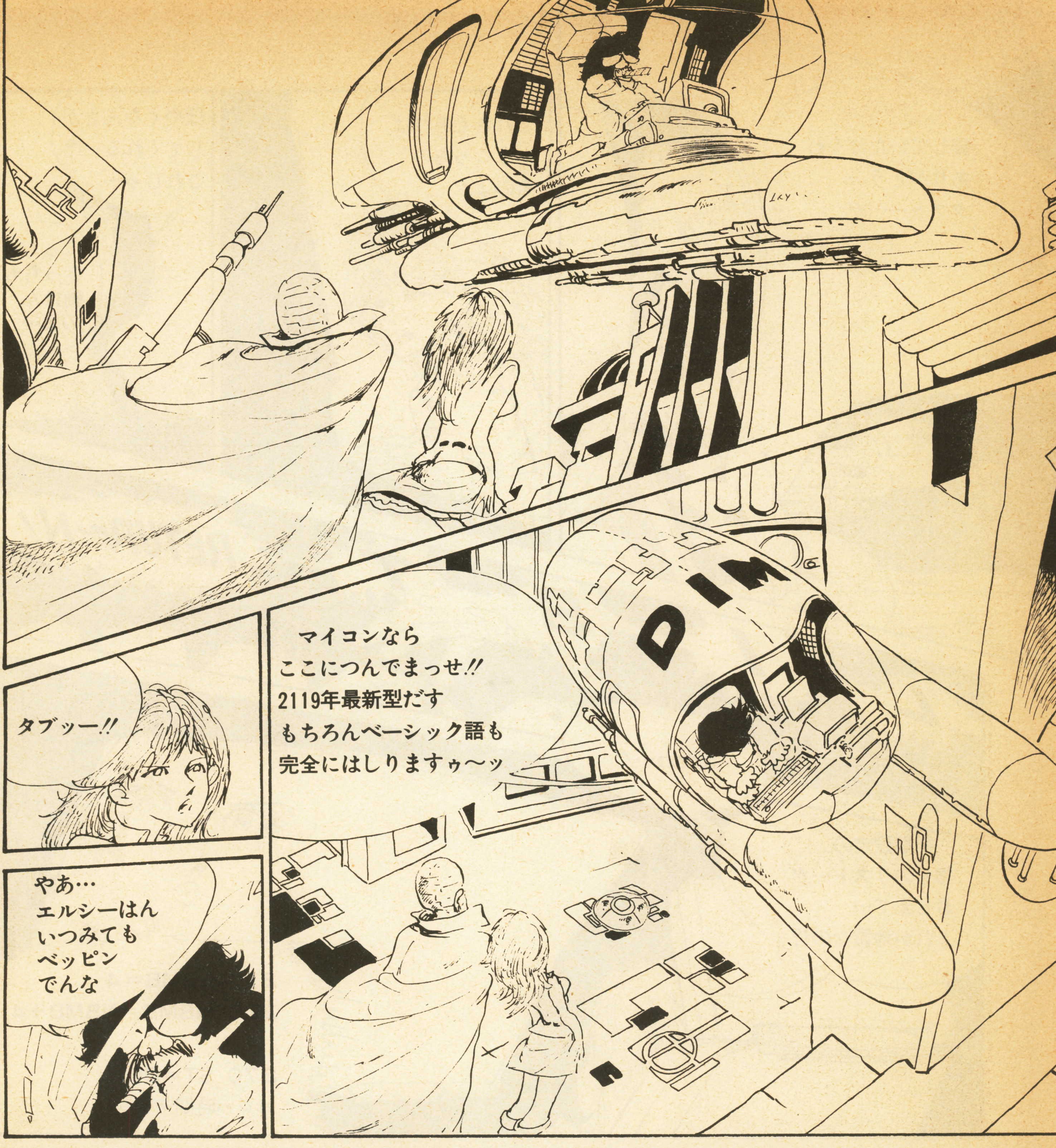
ここで指定した座標を中心に，囲まれた範囲を指定した色でぬるわけだ。

(870行) BEEP：ビープ／(警笛を)鳴らすという意味

《使い方》BEEP〔スイッチ〕*

スイッチを1にすると次のビープスイッチが0になるまで鳴り続ける。スイッチを省略した場合は，一瞬のあいだだけ鳴る。

* PC8001とMZ700にはPAINT文がない。リストの移植のポイント参照。



(880行) **GET@:ゲットアット／getは得るという意味**

《使い方》GET@（x_1，y_1）－（x_2，y_2），配列名

座標（x_1，y_1）－（x_2，y_2）を対角線とする範囲内の図形を配列名で指定した配列に覚えさせるんだ。

(910行) **PUT@:プット アット／putは置くという意味**

《使い方》PUT@（x_3，y_3）－（x_4，y_4），配列名

GET文で得た図形を(x_3，y_3）－（x_4，y_4)を対角線とする範囲に表示させる。

さて，これらの命令を念頭に置きながら……。

①まず足跡の形を何点か直線で囲ってそれらしく作る。……りんかくのポイント打ちをREAD−DATA文で，線をLINE文で 描く。

②その中をぬりつぶす。……PAINT文で……

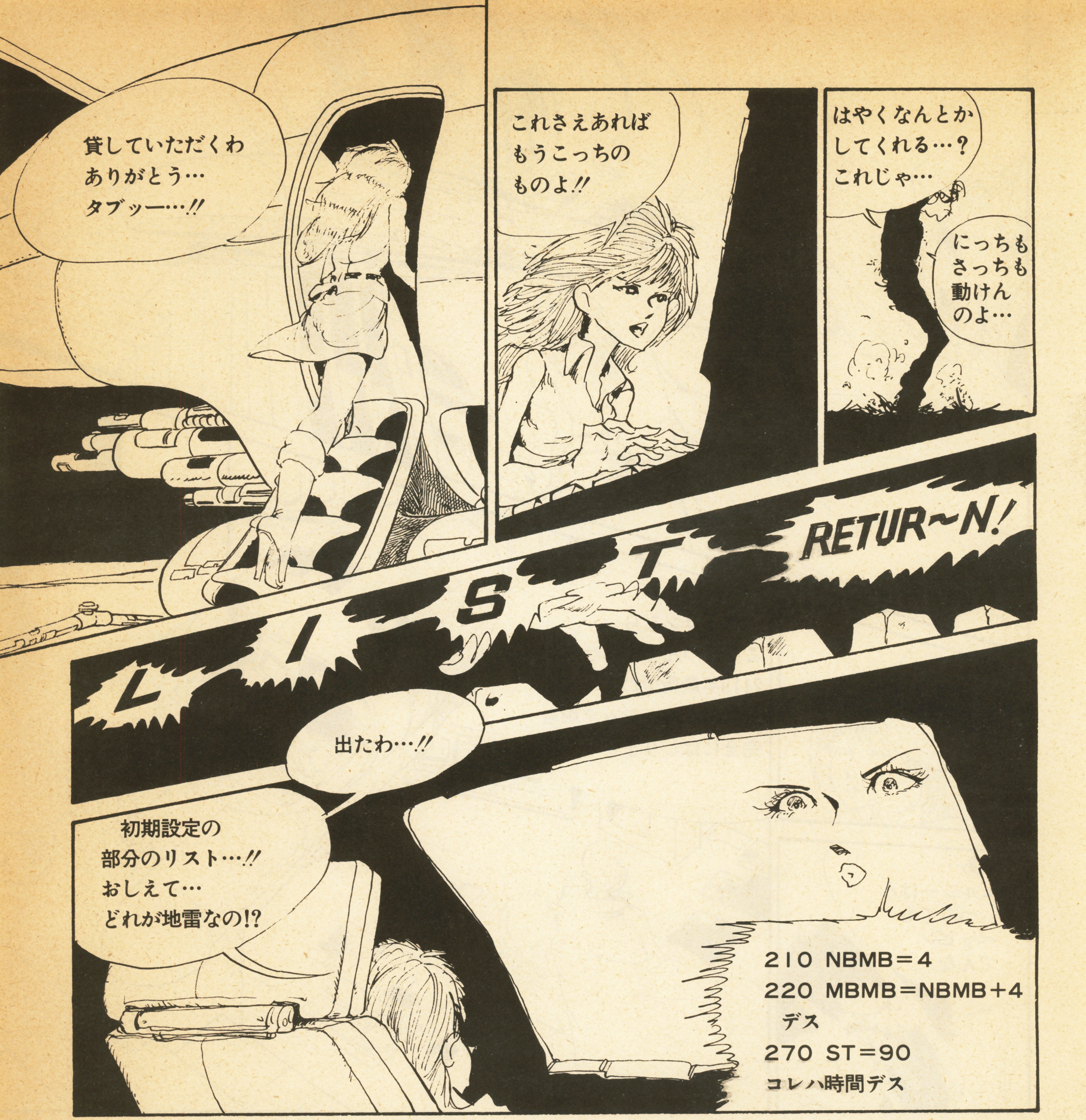
③一歩あるくたびにこんなことをやっていたのでは大変だから，一つ描いた足跡をGET文でパターンとして配列に覚えさせる。

④GET文で覚えた足跡をPUT文で描いてみる。これで2つ，左と右の足跡ができて都合がいい。ついでに一歩描くごとにBEEP文で音を出そう。

こんなものでどうだろうか？ あとは，足跡の色，表示する位置などを決める作業をすればいい。

この部分のプログラムのイメージはわいたかな？ リストと細かな解説については，74ページに掲載しているのでそれを参照しよう。

＊ BEEP文のスイッチは機種によってはないものがある



イニシャライズ* PART I DEFINT, CLSなど

さて，エルシーが初期設定のリストといっているね。これは，このゲームの中で使う変数や画面の設定などをあらかじめ決めておくところなんだ。ふつうプログラムを作るとき必ず必要になるところだ。そこで，アルゴのプログラムの初期設定をのぞくと……。

またまた見慣れない命令がでてきた。最初にまとめて整理しておこう。

(130行) **DEFINT：デフィント／DEFはDEFINE(定義する)の略。INTはINTEGER(整数の略)。**

《使い方》行番号　DEFINT　変数

130行では，A～Zが頭文字につく変数はすべて整数を代表するものだよ，と決めているわけだ。くわしいことは省略するが，こうやるとメモリが少し節約できて速くなる。

(150行) **CLS：クリア　スクリーン／clear (きれいにする), screen(画面)の省略形。画面を消去する。**

《使い方》CLS*

(160行) **WIDTH：ウィズ／幅という意味**

《使い方》WIDTH　1行にかける文字数，1画面の行数

(170行) **CONSOLE*：コンソール／制御卓という意味**

《使い方》CONSOLE　スクロール*開始行，スクロールする行数

＊　イニシャライズ＝初期設定のことだ。

＊　PC8001はCLSがないのでPRINT CHR$(12)：(プリントキャラクタダラー12)を使うと覚えておこう。



これで，110行から170行までは何をやろうとしているかわかるんじゃないかな。

190行からは定数をいろいろ決めているところだ。C0〜C7は色番号の指定。C2は2だから赤だね。

200行のNPSNは地雷原を越える人数だ。3人だね。

エルシーの捜す地雷の数は210行にNBMB＝4と出ている。そして，地雷は最高MBMB＝NBMB(＝4)＋4で8個まで増えることになっている。

そして270行のSTは制限タイムだ。

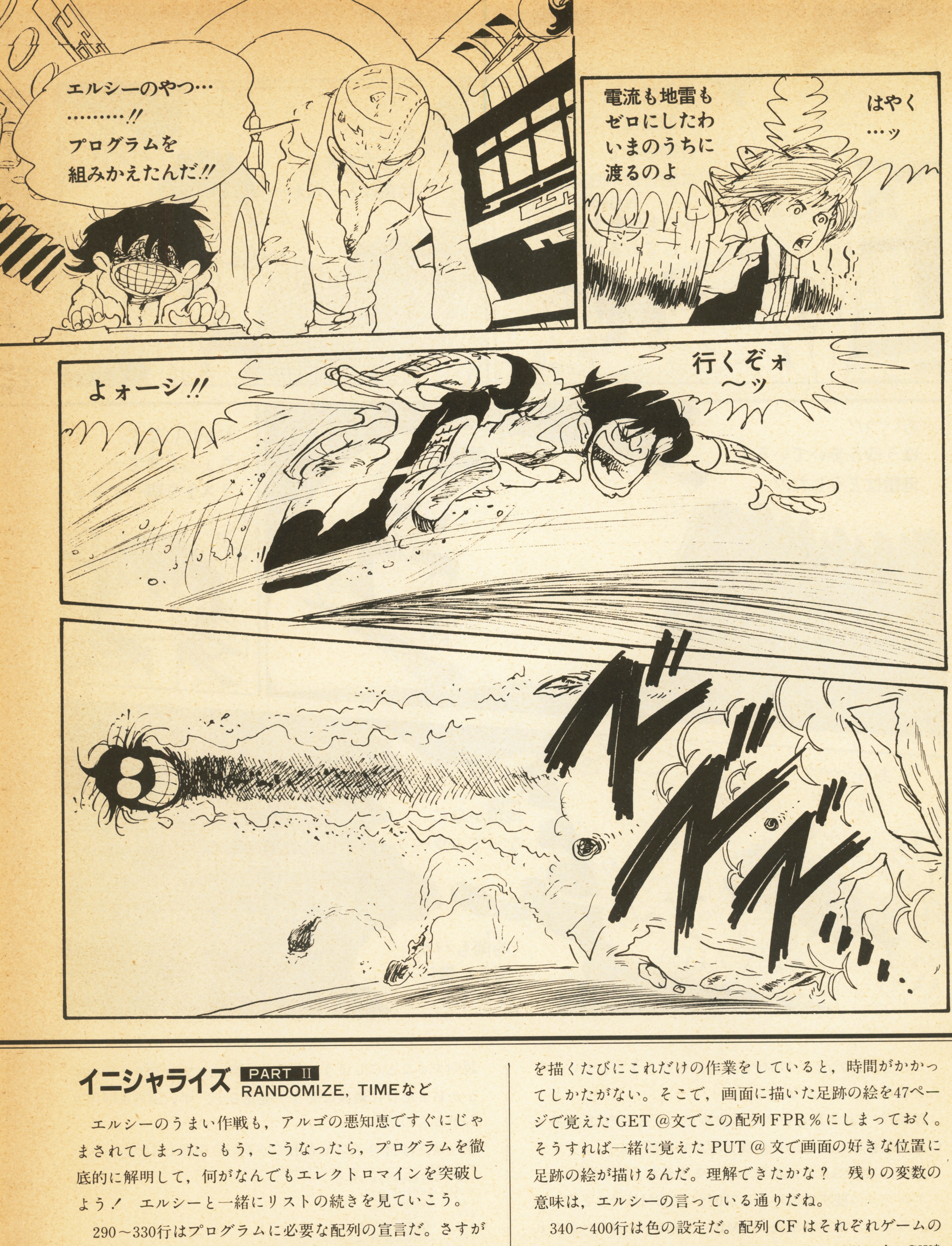
エルシーは，この地雷の数をゼロに，制限タイムはものすごくながくして突破しようとしたわけだね。

この他の部分はとりあえず今は必要ないから，また後で説明することにしよう。初期化の部分はプログラム全体にかかわるだけに最後にもう一度見ると，ずいぶんスッキリわかるものだ。

全リストは74ページから出ているから参照しよう。

```
110 REM ---------- Initialize ----------
120 'initialize FM-8
130   DEFINT A-Z
140   COLOR ,0
150   CLS
160   WIDTH 40,25
170   CONSOLE 0,25,0,0
180  set constant
190   C0=0 :C1=1 :C2=2 :C3=3 :C4=4 :C5=5 :C6=6 :C7=7
200   NPSN=3
210   NBMB=4
220   MBMB=NBMB+4
230   RB=12
240   MX=25 :MY=10
250   LX=MX*2 :LY=MY*2
260   DSZ=(((LX+1)*(LY+1)+7)/8+2-1)/2
270   ST=90
280   FALSE=0 :TRUE=NOT FALSE
```

＊ CONSOLEは機種によってさらに追加機能を持っている。
＊ スクロール＝巻き上げるという意味で，画面が上に上がることだ。



イニシャライズ PART II RANDOMIZE，TIMEなど

エルシーのうまい作戦も，アルゴの悪知恵ですぐにじゃまされてしまった。もう，こうなったら，プログラムを徹底的に解明して，何がなんでもエレクトロマインを突破しよう！ エルシーと一緒にリストの続きを見ていこう。

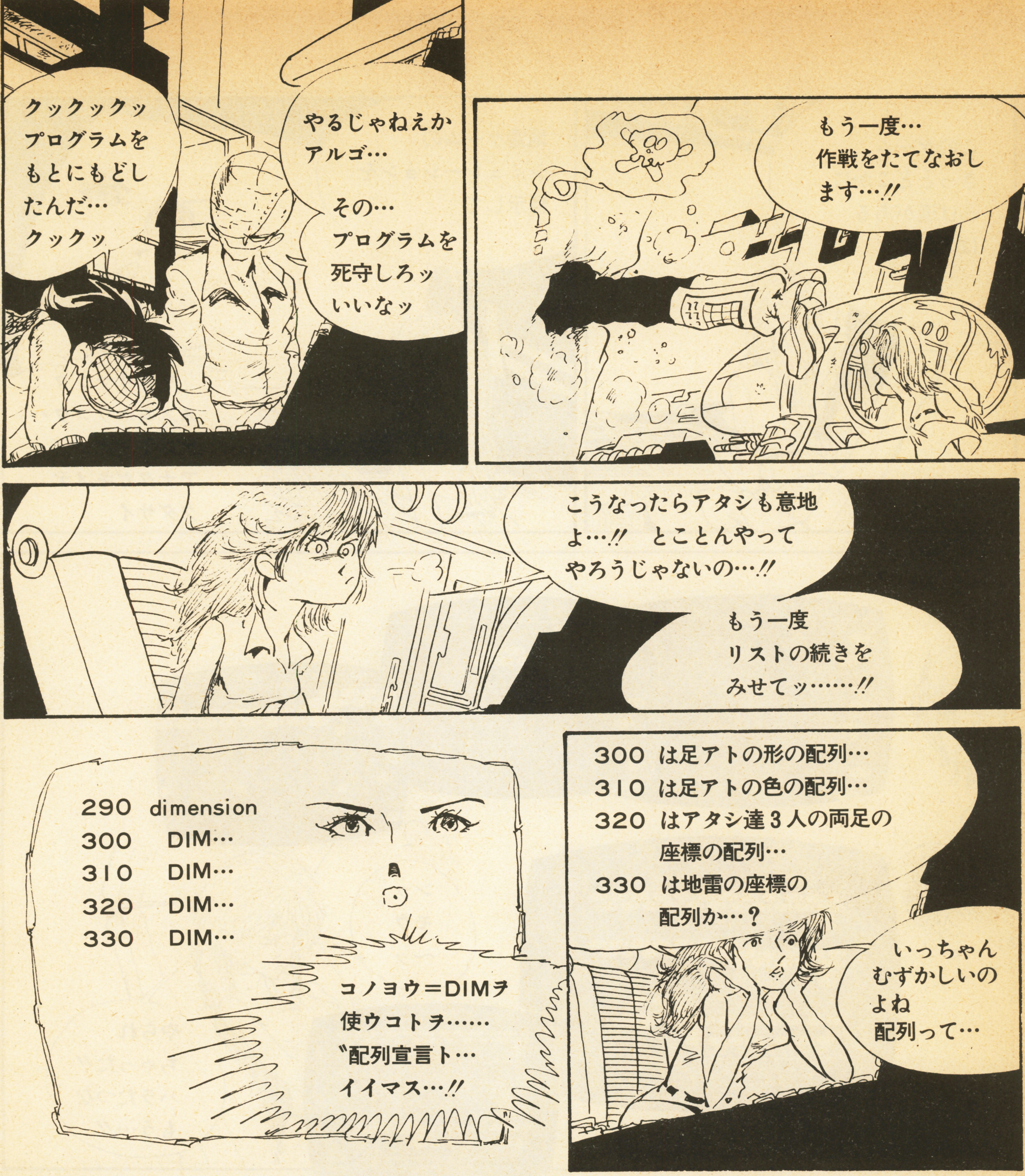
290〜330行はプログラムに必要な配列の宣言だ。さすがのエルシーも，何に使う配列なのかを理解するのに苦労しているようだね。このうちのFPR％はよくわかるように説明しておこう。これは足跡の形を覚えておくための配列だ。46〜47ページではあまり触れなかったけれど，リストの760〜860行では足跡の形を画面に描いている。画面に足跡を描くたびにこれだけの作業をしていると，時間がかかってしかたがない。そこで，画面に描いた足跡の絵を47ページで覚えたGET＠文でこの配列FPR％にしまっておく。そうすれば一緒に覚えたPUT＠文で画面の好きな位置に足跡の絵が描けるんだ。理解できたかな？ 残りの変数の意味は，エルシーの言っている通りだね。

340〜400行は色の設定だ。配列CFはそれぞれゲームの最初の画面に描かれる足跡と乱たち3人の足跡の色，CW*は壁の色，CB*は地雷の色を表している。

410〜440行には新しい命令が3つ出ている。

(420行) **RANDOMIZE：ランドマイズ／random（でたらめな），〜ize（〜化する）が複合した動詞。でたらめにする**

* CWはColor wall，CBはColor Bombあたりを連想して名付けたようだ。変数名はこうすると覚えやすいぞ！



という意味になる。

《使い方》行番号　RANDOMIZE　適当な数

(440行) DEF　FN：デファイン　ファンクション／define（定義する），function（関数）の省略形。英語を読んだ通り，関数を定義するという意味。

《使い方》行番号　DEF　FN関数名（変数*のリスト）＝関数の式（この式は自分で自由に作れる。利用者定義関数といい，自分でマイコンの新しい関数を作るわけだ。）

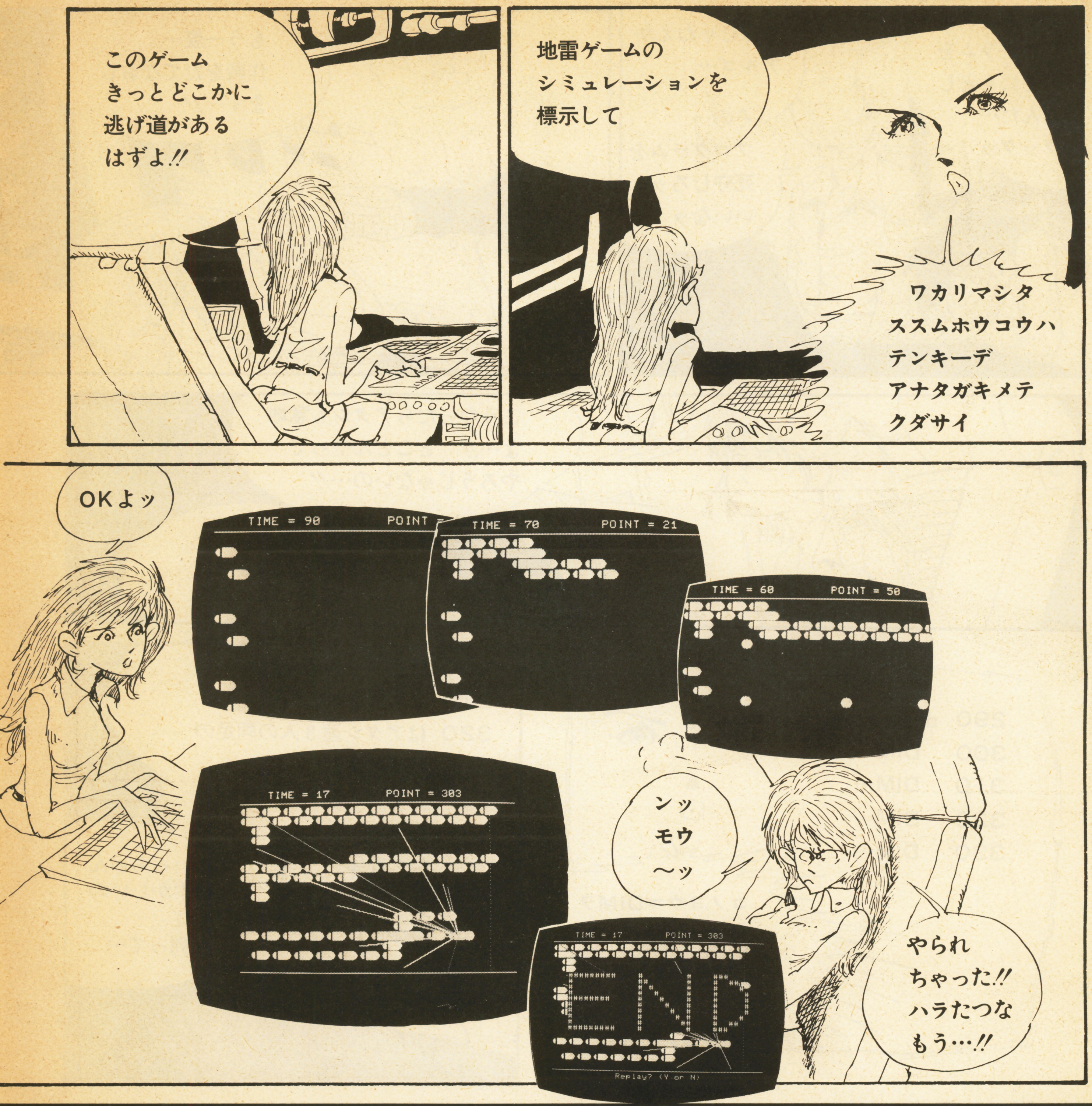
(440行) RND：ランダム／random の省略形。0～1の間の数字をでたらめに作り出す。

《使い方》行番号　RND（適当な数）

ベーシックには，呼び出すといつも違う値をとる RND という変わった関数がある。この関数に，よりでたらめな値を発生させるのを手伝っているのがこの RANDOMIZE だ。440行では，この RND 関数を使って関数を定義している。定義された関数 FNRAN は，1～SIZE までの整数のどれか1つを値にとる。

```
290 'dimension
300   DIM FPR%(DSZ)
310   DIM CF(NPSN)
320   DIM XFT(NPSN,2),YFT(NPSN,2)
330   DIM XBMB(MBMB),YBMB(MBMB)
340 'color
350   CF(0)=C6
360   CF(1)=C2
370   CF(2)=C6
380   CF(3)=C1
390   CW=C7
400   CB=C3
410 'randomize
420   RANDOMIZE(TIME/3)
430 'function
440   DEF FNRAN(SIZE)=INT(RND(1)*SIZE+1)
```

＊　変数…正確には「引数」（ひきすう）と言うんだけれど，今の段階では変数ということで覚えておいてくれ。



ゲームの舞台を作る 画面上に壁とゴールを描く

ここでは画面の作り方を説明しよう。まず，1130〜1180行を見てくれ。画面を消去してLINE文で上と下の壁を描いている。1160〜1180行は画面右側のゴールラインの表示だ。

PSET：ポイント　セット／point（点），set（置く）の省略。画面上の指定した位置に点を描く。

《使い方》PSET（x座標，y座標，点の色）

増やす値を4にしたFOR〜NEXT文とPSET文を組み合わせることによって，点線が描ける。

1190〜1230行はゲームの残り時間と得点の表示だ。

LOCATE：ロケイト／場所を決めるという意味

《使い方》LOCATE　水平位置，垂直位置

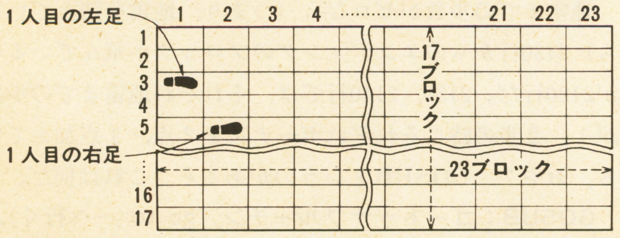
指定したキャラクタ座標にカーソルを移す。

これを使って残り時間は画面の1番上の行の左から5文字目，得点は同じ行の左から22文字目から書いている。

1250〜1320行は，3人の足の状態を設定している。このプログラムでは，53ページの図のように地雷原を23×17のブロックに分割して処理する。例として1270〜1280行の説明をしよう。これは，1人目の左足の位置を1番左の列の上から3行目のブロックに，1人目の右足の位置を左から2番目の列の上から5行目のブロックに決めているんだ。2人目と3人目の分は，この調子で理解してほしい。

1330〜1390行では，実際に足跡の絵を画面に描いている。





二重のFOR文は，外側のFOR文で3人の人間を，内側のFOR文で左足と右足を順に指定しているんだ。足跡の絵の表示にはおなじみのPUT@文を使っている。上の図で足の位置が入っている配列XFT，YFTの値を，グラフィック座標に変換しているからよく理解しよう。

```
1130   CLS
1140   LINE (0,12)-(639,12),PSET,CW
1150   LINE (0,188)-(639,188),PSET,CW
1160   FOR Y=13 TO 188 STEP 4
1170     PSET (23*MX+1,Y,CW)
1180   NEXT Y
1190 'point
1200   COLOR C7
1210   LOCATE 21,0 :PRINT "POINT =";PNT
1220   RT=ST
1230   LOCATE 4,0 :PRINT "TIME =";RT
1240 REM Footprint
1250 'initial condition
1260   PSN=1 :RL=2
1270   XFT(1,1)=1 :YFT(1,1)=3
1280   XFT(1,2)=2 :YFT(1,2)=5
1290   XFT(2,1)=1 :YFT(2,1)=9
1300   XFT(2,2)=2 :YFT(2,2)=11
1310   XFT(3,1)=1 :YFT(3,1)=15
1320   XFT(3,2)=2 :YFT(3,2)=17
1330 'put footprint
1340   FOR I=1 TO 3
1350    FOR J=1 TO 2
1360     X=XFT(I,J)*MX :Y=YFT(I,J)*MY
1370     PUT@ (X-MX,Y-MY)-(X+MX,Y+MY),FPR%,PSET,CF(I)
1380    NEXT J
1390   NEXT I
```



地雷に近づくと音を出す GOSUB, RETURN

地雷原をガムシャラに進んだのでは，命がいくつあっても足りない。そこで，エルシーは，地雷に近づくと警報が鳴るようにプログラムを改良した。

リストの2080〜2150行では，地雷原を渡る人間の現在位置と地雷の位置との距離を求め，それに従って得点を増やしている。まず2100行は距離の計算だ。現在位置から何歩歩くと地雷を踏むかという歩数を出している。

ABS：アブソリュート／absolute（絶対的な）。数学のabsolute value（絶対値）からきている。

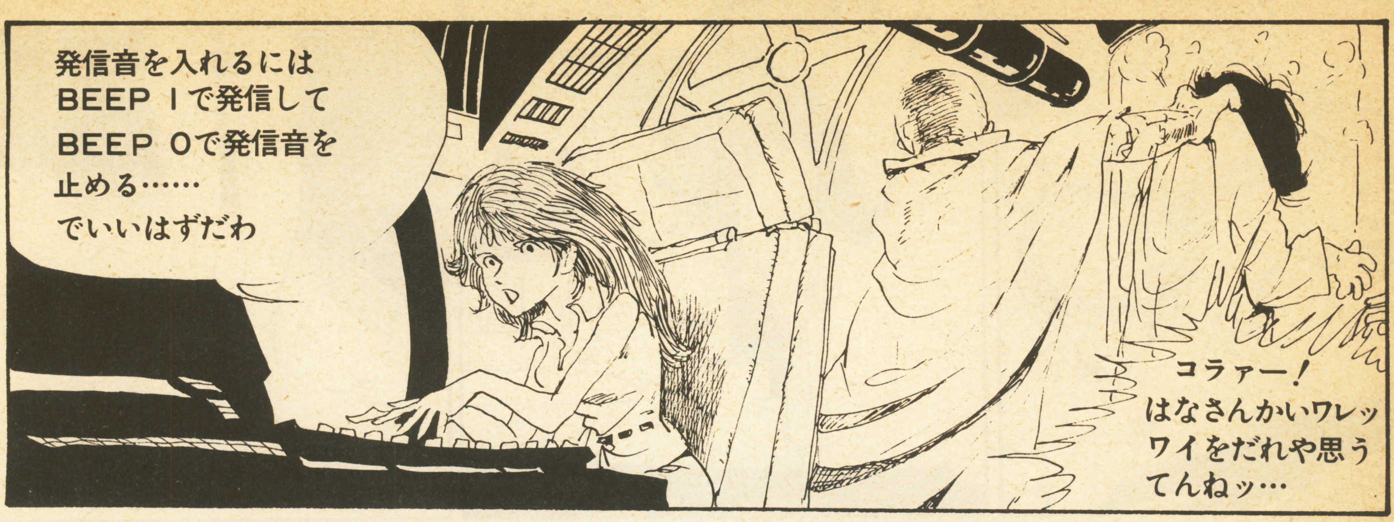
《使い方》ABS（数） 負の数を正の数にする関数。

地雷までの歩数がゼロなら（つまり，地雷を踏んでいたら）2430行からのエンディングのブロックへ飛んでいるのが2120行だ。2130，2140行では，それぞれ地雷までの歩数が1，2歩の時にそれぞれボーナスを2点，1点加えている。おや？ GOSUBというのがあるぞ。これは何だ。

GOSUB：ゴートゥサブルーチン／go to（〜へ行く），subroutine（subは補助，routineはおきまりの仕事）

《使い方》GOSUB サブルーチンの先頭行番号。

同じ処理を何度もくり返す場合に，その処理をプログラムの中でサブルーチンとして書いておけば，何度も使うことができる。サブルーチンの最後にはRETURN文を書いておいて（リストの3200〜3250行），このGOSUB文で呼び



発信音を出すサブルーチン

```
2130 IF PBD=1 THEN
     GOSUB 3200
     :GOSUB 3200
2140 IF PBD=2 THEN
     GOSUB 3200
……………………
3200 REM…Point…
3210 'Point increment
3220 BEEP 1:BEEP 0
……………………
```

地雷までの距離はPBDだから，距離１にまで接近したら，BEEP音を出すサブルーチンへ２回行き，距離２くらいのときはBEEPの警告音は１回にしておこう…



出せばいい。2130行のようにGOSUB 3200としておけば，3200～3250行のサブルーチンを実行した点，GOSUB文の次の文(ここでは２つ目のGOSUB文)にもどってくる。つまり，GOSUB 3200と書いておけば得点が１点増えると考えればいいんだ。2130行には２回書いてあるから3200行のサブルーチンへ２回行って２点増えるね。

それでは，3200～3250行の得点のサブルーチンを見てみよう。3230行で得点を増やして，3240行で新しい得点を画面に書いている，3220行がエルシーの付け加えた部分だ。どうしてここにBEEP文を入れたのか理解できるかな？

エルシーの考えはこうだ。この得点を加算するサブルーチンは，１歩前進すると１回（これは2070行なので後でもう一度触れる)，前進した位置が地雷から２歩の場所だったら２回，地雷から１歩の場所だったら３回呼ばれる。だからこのサブルーチンにBEEP文を入れておけば，危険であればあるほど警報が多く鳴る。要するに，危険な場所を通れば得点が多く加算されるのをうまく使っているんだ。

```
2080 'person-bomb distance
2090   FOR I=1 TO NBMB
2100    PBD=ABS((XBMB(I)-XFT(PSN,RL)))¥2+ABS(YBMB(I)-YFT(PSN,RL))
2110    IF PBD>2 GOTO 2150
2120     IF PBD=0 THEN ET=1:GOTO 2430:'Ending
2130     IF PBD=1 THEN GOSUB 3200:GOSUB 3200:'Point*2
2140     IF PBD=2 THEN GOSUB 3200:'Point
2150   NEXT I

3200 REM ---------- Point ----------
3210 'point increment
3220   BEEP 1 :BEEP 0
3230   PNT=PNT+1
3240   LOCATE 28,0 :PRINT PNT;
3250  RETURN
```



歩みはじめ 実際のゲームの遊び方

エルシーによるプログラムの改良は役に立ったようだね。彼女は，みごとに地雷原を渡りきってしまった。地雷原を1歩1歩前進する動作を，これから見ていくことにしよう。

1620～1680行は画面に「START」というメッセージを書いてブザーを鳴らし，出発の合図を送っている。1640行にはINKEY＄という新しい関数が出ているね。これの説明は68ページでするけれど，D＄=INKEY＄と書いておけば今押されているキーの文字が変数D＄に代入されると覚えておいてくれ。つまり，ゲームで遊ぶ人間の要求はこの関数を使って伝わっているんだ。

1700行からは歩みの処理が続いているのだけれど，ここの説明は後まわしにして，まずゲームの遊び方を述べよう。

画面上で上・下・左・右に進みたい時は，テンキーのそれぞれ「8」・「2」・「4」・「6」を押す。基本的にはこの4つがあればいい。しかし，このプログラムでは，ゲームを途中でやめたくなった時のために自爆ができるようになっている。「Q」のキーがその指令だ。これらのキーのレイアウトをまとめたものが右のページに示してある。

さあ，それでは歩みの処理を見ていこう。1720～1750行は，キーの入力待ちだ。上で決めた5つのキーのどれかが押されるまで，残り時間を減らしながら待っている。何もキーが押されないか，押されてもゲームに関係のないキー



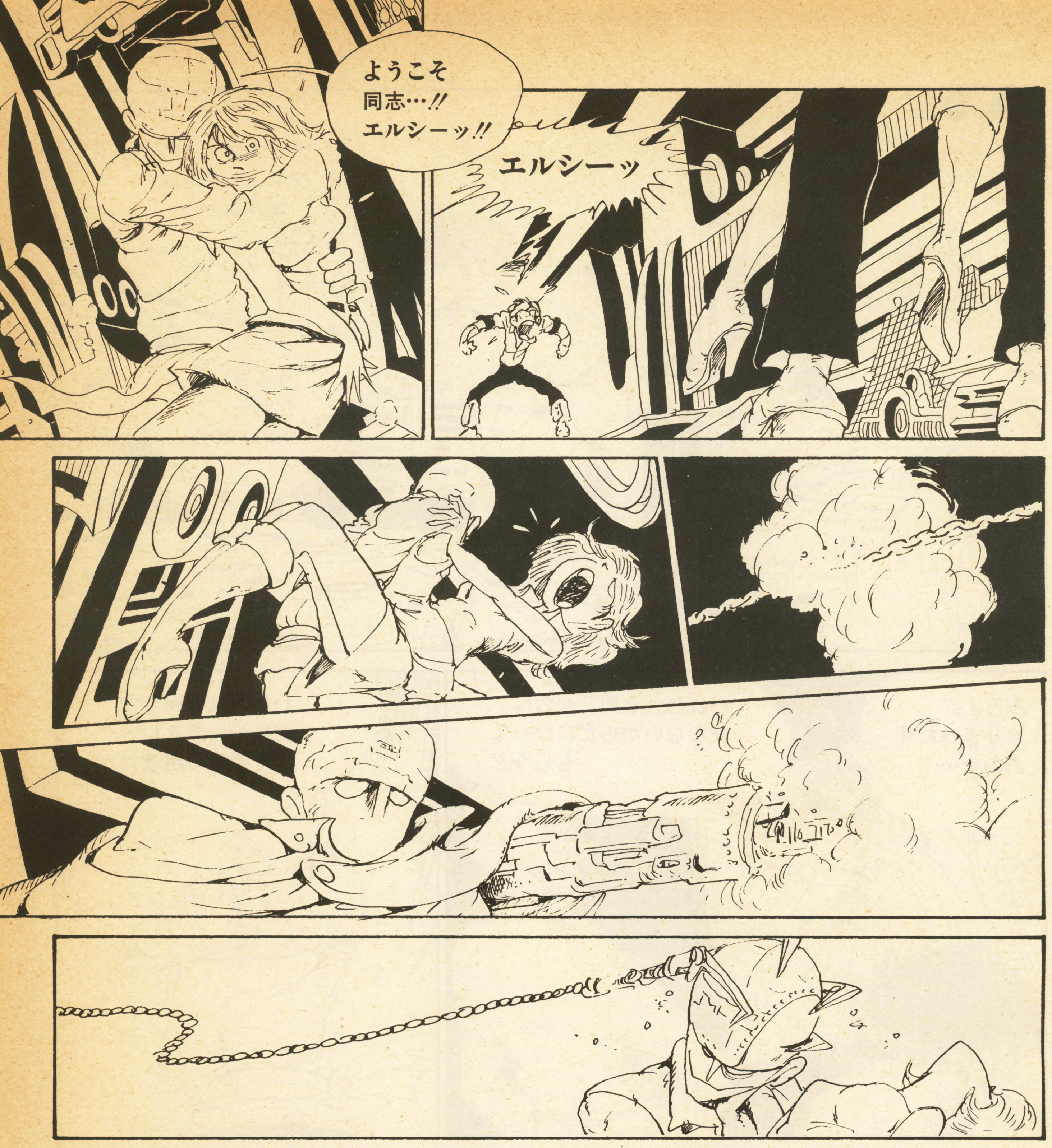
エレクトロマインを
遊ぶための
キー・リスト

キー	動作
8	上に進む
4	左に進む
6	右に進む
2	下に進む
Q	自爆する

（中央：現在位置）

であれば1730〜1750行をまわり続ける。1730行を1回通るたびに変数TCの値は1ずつ増えるね。これが20を越えると3270行からのサブルーチンが呼ばれる。このサブルーチンは後で説明するけれど，残り時間を減らして時間切れになったかどうかをチェックしているんだ。1740行のINKEY$関数で読み込んだキー入力が妥当なものであれば，1750行を通過して実際の歩みの処理に進んでいく。

```
1620 REM ---------- Start ----------
1630 'start!
1640   D$=INKEY$
1650   LOCATE 14,24 :COLOR CF(PSN) :PRINT "S T A R T";
1660   FOR I=1 TO 180 :BEEP 1 :BEEP 0 :NEXT I
1670   LOCATE 14,24 :PRINT "         ";
1680   COLOR C7
1690 '
1700 REM ---------- Walk ----------
1710 REM Input
1720 'get character  'add counter
1730   TC=TC+1 :IF TC>20 THEN TC=0:GOSUB 3270:'Time
1740   D$=INKEY$
1750   IF D$<>"2" AND D$<>"4" AND D$<>"6" AND D$<>"8"
        AND D$<>"Q" GOTO 1730
1760   TC=TC+20
```



歩みの処理 VAL ON GOTO

乱はあわてて飛び出していったけど，案の定すぐに行く手をはばまれて引き返してきた。「急がばまわれ」。ここは腰を据えて，歩みの処理の続きをながめていこう。

押されたキーが「Q」(大文字しか受け付けないので注意！）なら，2430行からのエンディングに飛んでいる。変数ETを3にしているのは爆発の表示を指定しているのだけど，この説明は後でする。

1790行に新しい関数が出ているので説明しておく。

VAL：バリュー／value（価値）

《使い方》VAL（文字列）

文字列の表す数値を値にとる。たとえば“2”なら2。

この関数で，キーボードから入力された“2”，“4”，“6”，“8”の文字*を2，4，6，8の数値に変換する。

1800行でゼロに設定している変数DX，DYはそれぞれ横方向，縦方向の位置の変化分を表すものだ。たとえば，下に1歩進むのならば，DYを1にすればいい。

1810行にON GOTO文が出てきた。説明しよう。

ON GOTO：オン ゴートゥ／on～（～に基づいて），go to～（～へ行く）。条件により，飛び方を決める。

《使い方》ON 式 GOTO 行番号①，行番号②，…

式の値が1ならば行番号①，2ならば行番号②，…に飛ぶ。

進む方向は「2」，「4」，「6」，「8」のキーで入力される

＊ 6とか8といっても数字として扱う場合と，文字として扱う場合があるんだよ。どちらの場合もたし算ができるんだ(↗)(↘)けど文字としてのたし算はPRINT“6”＋“8”→“68”となるんだ。数字の場合は14だね。違いをよ～く理解しよう。



ので，これをＶＡＬ関数で数値に変えた後で２でわってやれば１，２，３，４の数になる。これを利用して，飛び先を決めているのが1810行なんだ。

1820～1970行は入力されたキーによる上・下・左・右の移動の処理を担当している。1820～1850行を例にとろう。

1820行に飛んでくるということは，1810行のＤ￥*2の値が１になったことを表している。ということは，「２」のキーが押されたんだね。これは下への移動の指定だ。前に説明した通り，下に移動するには変数ＤＹの値を１にすればいい。これは1830行で処理してあるね。1840行は少しわかりにくいが，左右どちらの足を踏み出すかの決定だ。下に動きたいのだから，股を開いていれば（左右の足が上下方向に離れていれば）左足を，そうでなければ右足を１歩下に進めればよい。1860～1970行もこの部分と同様だ。

```
1770 'set direction and right-left
1780   IF D$="Q" THEN ET=3:GOTO 2430:'Ending
1790   D=VAL(D$)
1800   DX=0 :DY=0
1810   ON D¥2 GOTO 1820,1860,1900,1940
1820   REM D=2...down
1830     DY=1
1840     IF YFT(PSN,1)<YFT(PSN,2)-1 THEN RL=1 ELSE RL=2
1850     GOTO 1980
1860   REM D=4...left
1870     DX=-2
1880     IF XFT(PSN,1)<XFT(PSN,2) THEN RL=2 ELSE RL=1
1890     GOTO 1980
1900   REM D=6...right
1910     DX=2
1920     IF XFT(PSN,1)<XFT(PSN,2) THEN RL=1 ELSE RL=2
1930     GOTO 1980
1940   REM D=8...up
1950     DY=-1
1960     IF YFT(PSN,1)<YFT(PSN,2)-1 THEN RL=2 ELSE RL=1
1970     GOTO 1980
```

＊　￥…整数型のわり算。商の小数点以下は切り捨てる。たとえば，10￥３＝３，１￥３＝０。33ページを参照。



同じ場所は通れない　キャラクタ座標 グラフィック座標

エルシーの通った跡をたどって渡れば地雷は踏まない!そう考えた乱はさっそく実行に移したが，安全なはずの道にはナイフが待っていた!!

同じ道は通れないようにする方法も含めて，歩みの処理を続けて見ていくことにしよう。

2000〜2010行は画面からはみ出さないかどうかのチェックだ。前のページで説明した変数DX，DYの値を現在位置に加えてみて，歩ける領域からはみ出していたら1700行に戻って入力のやり直しをするんだ。

同じ場所が，二度と通れないようになっている秘密は，2020〜2030行だ。ここから後を理解してもらうために，まず座標の話をしておかなくてはいけない。

マイコンの画面には縦に20か25行*1，横に40か80桁*2の文字が書ける。これを使って文字の位置が指定できるんだ。たとえば，図1でAの文字は上から2番目の行の左から5桁目の位置にあるといえるね。マイコンでは行も桁も0番からかぞえはじめるので，Aの位置は1番の行の4番の桁ということになる。これは（4，1）という数字の組で表すとわかりやすい(桁の番号が最初だから注意)。そこで，Aはキャラクタ座標（4，1）にあるという表現を使うんだ。

マイコンには，これとは別にグラフィック座標というものがある。これは画面に絵を書いたりするためのもので，

*1，*2　この数字はマイコンによって違うので注意!



図1．
キャラクタ座標

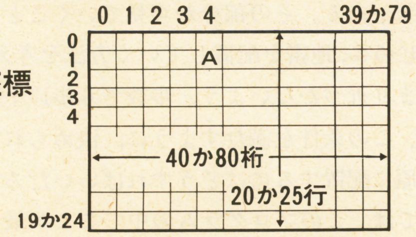
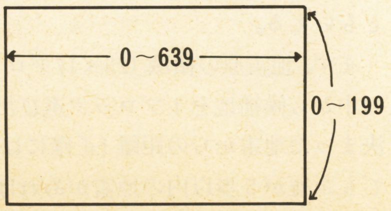


図2．
グラフィック座標



横640，縦200の点が指定可能だ。たとえば画面左上隅の点は，グラフィック座標（0，0）の位置だといえるね。

POINT：ポイント／点という意味がある。

《使い方》POINT（*x*，*y*）　座標*（*x*，*y*）に点が描いてあれば－1，そうでなければ0を値とする。

話をもとに戻そう。一度通ったことのある場所は，画面に足跡が描いてあるはずだね。そこで，これから行こうとしている場所のグラフィック座標を2020行で計算して，その場所に点が描いてあるかを2030行でチェックしているんだ。もし点があれば，1700行にもどってキーの入力をし直す。

```
1980 REM Step
1990 'step?
2000   XT=XFT(PSN,RL)+DX  :YT=YFT(PSN,RL)+DY
2010    IF YT<=1 OR YT=>19 OR XT<=0 GOTO 1700:'......Walk
2020   X=XT*MX  :Y=YT*MY
2030    IF POINT(X,Y) GOTO 1700:'..................Walk
```

＊ 単に座標とあればグラフィック座標だと考えること。



地雷が見える! PART II 地雷の配置

エルシーに負けじと，乱もプログラムの改良を始めたぞ。なんと，乱は地雷を目に見えるようにしてしまった。座標の説明が入ったせいで解説できなかった歩みの処理のまとめと地雷を画面に表示させる処理の話をしよう。

63ページの2040〜2070行を見てくれ。2050行では決定した新しい足の位置を配列にしまっている。足跡の絵を画面に描いているのが2060行だ。このPUT@文は前に説明したね。無事に1歩進めたら2070行で得点の加算サブルーチンへ飛ぶ。

さて，次は乱が作った地雷の表示プログラムを考えていこう。そのためには，まず，地雷を配置する処理を理解する必要がある。その部分から見ていくことにする。

はじめに，地雷を配置していく方法を考える。地雷どうしがあまり近づかないよう，歩数で6歩以上離れるようにしよう。この条件を満たすように，決められた個数の地雷を地雷原に配置するにはどうすればいいだろうか？　考え方としては，このプログラムの中でここが最も難しいところともいえる。

まず，地雷原の領域（23×17ブロックの大きさだったね）の中から候補地を1ブロック選び出す。すでに配置位置の決まった地雷からの距離（正確には歩数）を計算して，1つでも距離が5歩以内の地雷があれば，その候補地は失格となる。すべての地雷からの距離が十分にあればその候補地



は合格である。こうして，決められた個数だけ地雷を配置すれば終わりだ。

実際のプログラムを見ていこう。1410〜1560行がそれだ。1440，1450行で候補地を決めている。440行で定義した関数FNRANを使ってるね。1460〜1520行は他の地雷から近すぎないかどうかのチェックだ。ここはくわしく説明すると長くなるので，前に述べた考え方と照らし合わせて理解してほしい。

地雷の配置の方法はわかってもらえただろうか。こういうたぐいの考え方は，また次の機会にでも説明したいと思う。アルゴが頭をひねって生み出したこのプログラムに，乱は1550行を付け加えた。このCIRCLE命令で乱は地雷の配置した位置に円を描いて見えるようにしたわけだ（次ページ参照）。

```
1400 REM Mine
1410 'lay mines for free space at random
1420   FOR I=1 TO NBMB
1430    REM serch
1440     XBMB(I)=FNRAN(19)+3
1450     YBMB(I)=FNRAN(17)+1
1460     BRY=TRUE
1470     J=1
1480     IF BRY AND J<I THEN 1490 ELSE 1530
1490      BBD=ABS(XBMB(I)-XBMB(J))+ABS(YBMB(I)-YBMB(J))
1500      IF BBD<=5 THEN BRY=FALSE
1510      J=J+1
1520     GOTO 1480
1530    IF NOT BRY GOTO 1430:'serch
1540    FOR J=1 TO 10 :BEEP 1 :BEEP 0 :NEXT J
1550    CIRCLE (XBMB(I)*MX,YBMB(I)*MY),RB,CB,,,,F,PSET
1560   NEXT I

2040 'step
2050   XFT(PSN,RL)=XT :YFT(PSN,RL)=YT
2060   PUT@ (X-MX,Y-MY)-(X+MX,Y+MY),FPR%,PSET,CF(PSN)
2070   GOSUB 3200:'Point
```

```
1400 REM Mine
1410 ´lay mines for free space at random
1420   FOR I=1  TO NBMB
1430    REM search
1440     XBMB(I)=FNRAN(19)+3
1450     YBMB(I)=FNRAN(17)+1
1460     BRY=TRUE
1470     J=1
1480     IF BRY AND J<I THEN 1490
         ELSE 1530
1490      BBD=ABS(XBMB(I)-XBMB(J))
          +ABS(YBMB(I)-YBMB(J))
1500      IF BBD<=5 THEN BRY=FALSE
1510      J=J+1
1520     GOTO 1480
1530    IF NOT BRY GOTO 1430:´search
1540    FOR J=1 TO 10:BEEP 1:BEEP 0
        :NEXT J
1550    CIRCLE(XBMB(I)*MX,YBMB(I)*MY)
        ,RB,CB,,,,F,PSET
1560    NEXT I
```



地雷が見える! PART II CIRCLE文を使う

乱が組み込んだプログラムの正体がわかった。アルゴが地雷の位置設定が的確であるかどうかを必死になって，いくつものチェックを重ねたのも，これで水の泡。乱数なんていうものまで持ち出してきたのだから気の毒だ。

もう，わかっただろう。乱がどんなプログラムを付け加えたかが。地雷の位置がわかるように表示させるプログラムなんだ。アルゴとキルが，敵ながらあっぱれと感心している姿が見えるだろう。

地雷は画面上では，円を描くことによって示すとしよう。円を描くのは，CIRCLE文だ。（前ページのリスト参照）

（1550行）CIRCLE：サークル／円という意味だ。

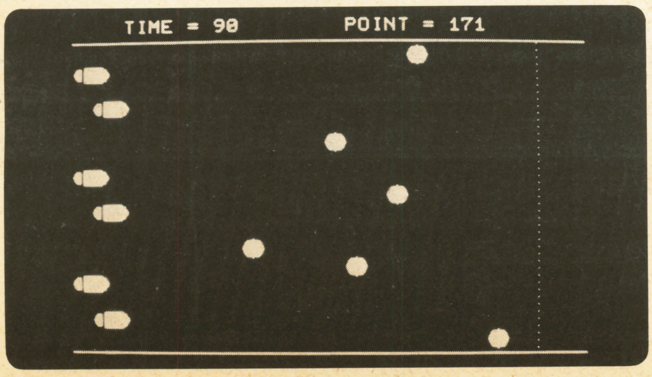
《使い方》行番号　CIRCLE（x座標，y座標），半径，カラーコード，比率，開始位置，終了位置，{F/N}，機能

1440〜1450行で決めた地雷のx座標，y座標が地雷を描くときの中心座標になる。そして，地雷の半径，色をRB，CBで設定。地雷を塗りつぶしたいときは{F/N}のFを指定してやればよい。Nを指定したり，省略すると塗りつぶされないよ。機能はここではPSETを使う。比率，開始・終了位置は，ここでは不要だね。

1550行で地雷を描く。乱の作ったプログラムの威力を見せてやろう。1420行と1560行のFOR〜NEXT文で，表示すべきすべての地雷を描くんだ。そして，地雷の表示と同



時にブザー音を鳴らしている。1540行だ。BEEP文で，ブザー音を出すんだったね。音の長さは，FOR～NEXT 文で調節するテクニックを使っている。



ランダムは，すでにゴールインしていて，乱をいまや遅しと待っているね。ここで，ゴールインしたかどうかのチェックをするとしようか。2160～2180行だ。

2180行の23というのは，ブロック数だ。ここで，ちょっと53ページを見てほしい。地雷原を画面では，横23，縦17のブロックに分けて，計算しやすいようにしたね。このゲームでは，左足が23ブロックを越えた時をゴールインとしているんだ。23未満であれば，乱と同様に続けて地雷原を歩かなくてはいけない。この判定は IF 文だ。XFT( ) は，その時点での足跡の x 座標だ。

```
2160 REM Reach
2170 'reach?
2180   IF XFT(PSN,RL)<23 GOTO 1700:'................Walk
```



地雷が消えた! PRESET

乱が地雷原をノンビリと歩いていたために，乱がゴールインする前にすべての地雷の姿が見えなくなってしまった。

ランダムの記憶をたよりに，乱は果たしてこの地雷原をうまく通り抜けられるだろうか。

リストの1580～1600行では，すべての地雷を消す作業をしている。ここでは，地雷だけを消すのだから，ＣＬＳ文は使えない。だって，画面がみんな消えてしまうからね。

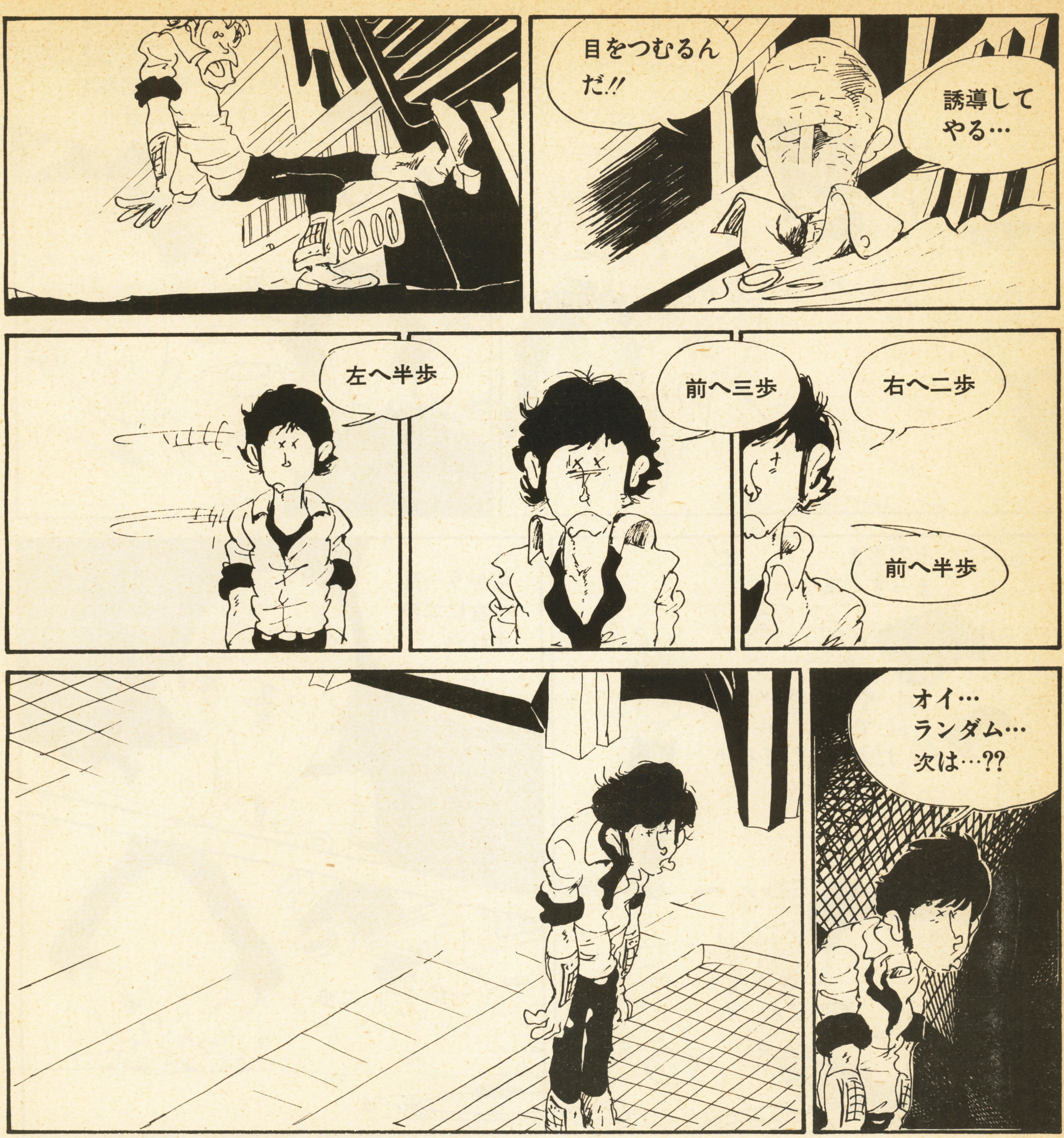
こういう場合は，描かれた地雷の色を，画面の背景色と同じにしてしまえばいい。こうすると結果的には地雷が消えたことになるわけだ。

PRESET：**ポイントリセット／point（点），reset(置き直す）の省略形。画面上の指定した位置の点を消去。**

CIRCLE文の中の一つの機能としてPRESETを使って地雷を消している。FOR文で，画面に表示されているすべての地雷に対して行うんだ。

エルシー，ランダムの２人がゴールインしているね。最後は乱だ！　乱が地雷原の突破に成功しなくては，もともこもないのだから，とにかくガンバってもらわなくてはならない。乱を見ていると冷や汗ものだけれど。

誰か１人，地雷原の通り抜けに成功するたびに，20点得点するようにしよう。2200行だ。続けて，次の人間が挑戦。挑戦者は３人までと決まっている。３人というのは2220行



のNPSNにあたり，この行では，何人通過成功したのかを判定している。３人全員が通過したら，GOTO文で次の処理へ飛ぼう。でも，仮に３人が成功したからといって安心はできない。

なぜかって？　次の地雷原が待っているからだ。

新たな挑戦者に対しても，また地雷の位置を教えてくれる。そのために，2240～2270行で，地雷を画面に表示する。地雷音も一緒だ。地雷は円で描くのだから，CIRCLE文を使うんだったね。もちろん地雷の色は紫色でぬりつぶすのだからCIRCLE文にはFの設定が必要だよ。

そして，乱が，地雷が消えてあわてたように，また地雷を消してしまおう。66ページでやったように，PRESETを使えばいいんだ。FOR～NEXT文で，すべての地雷を消し忘れないようにね。2280～2310行だ。それが終わったら，スタートだ。

```
1570 'hide mines
1580   FOR I=1 TO NBMB
1590    CIRCLE (XBMB(I)*MX,YBMB(I)*MY),RB,CB,,,,F,PRESET
1600   NEXT I
1610 '

2190 'next person or next screen
2200   FOR I=1 TO 20 :GOSUB 3200 :NEXT I :'Point*20
2210   PSN=PSN+1
2220   IF PSN>NPSN GOTO 2320
2230   REM next person
2240     FOR I=1 TO NBMB
2250      FOR J=1 TO 10 :BEEP 1 :BEEP 0 :NEXT J
2260      CIRCLE (XBMB(I)*MX,YBMB(I)*MY),RB,CB,,,,F,PSET
2270     NEXT I
2280     FOR I=1 TO NBMB
2290      CIRCLE (XBMB(I)*MX,YBMB(I)*MY),RB,CB,,,,F,PRESET
2300     NEXT I
2310     GOTO 1620:'.............................Start
```



突破成功! 入力待ちに使う INKEY$

ここで，残り時間のカウントをしてみよう。3270～3340行が，残り時間を計算するサブルーチンだ。残り時間の変数名はRTだったね。残り時間を知らせるわけだから，もちろん，RTは減っていくわけだ。減る割合を1としよう。3290行だ。計算が終わったら，画面で教えてあげるんだ。画面の位置設定は，LOCATE 文だったね。3300行で，画面の位置は，1番上の行の左から11文字目からであることがわかる。確認したら，PRINT 文で表示しよう。

もし，RTが10より大のときは，そのままRETURN 文で戻る。RTが9以下のときには，ブザー音を鳴らして教えてあげよう。RT＝0のときは，時間切れということだ。ゲームは終了する。このRTの判定は3310～3330行。

さて，次は成功シーンを作ろう。

2910行は，この行が実行されるまでにあったキーボードからの入力をすべて意味のない入力とするためのものだ。

INKEY$：インキーダラー／ in（～の中に），key（keyboardの略）。キーボードからの文字入力を意味する。

《使い方》キーボードからの入力があればその文字を，な

```
3270 REM ---------- Time ----------
3280 'time decrement
3290   RT=RT-1
3300   LOCATE 10,0 :PRINT RT;
3310   IF RT>=10 GOTO 3340
3320     IF RT=0 THEN ET=4:RETURN 2430:'Ending
3330     IF RT<=9 THEN BEEP
3340 RETURN
```

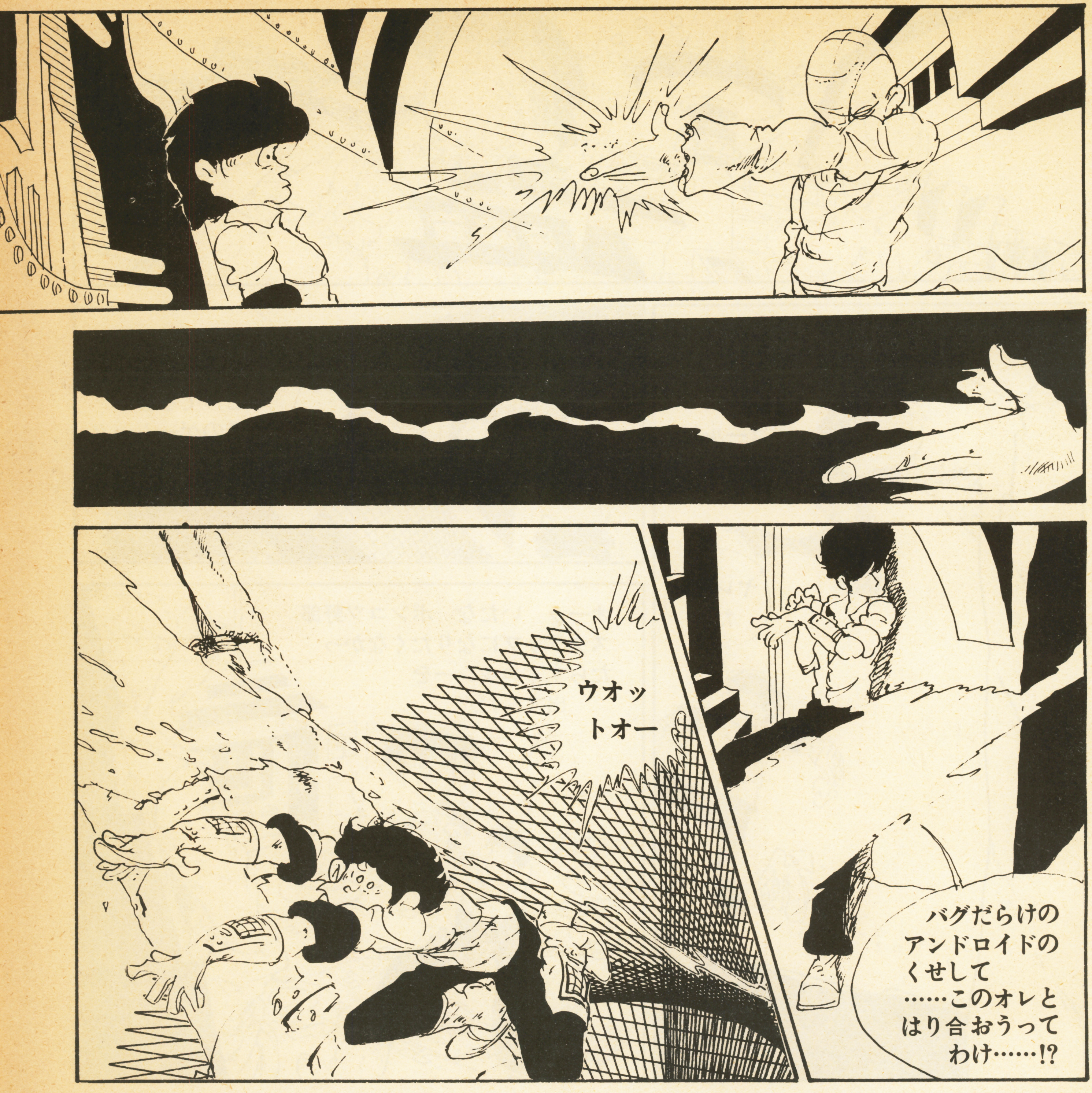


ければ，空(カラ)文字を与える。

2930～3070行は43ページでやったテクニックの応用だね。LOCATE, PRINT文をうまく使ってキャラクタによる"突破成功"の表示だ。

3100～3140行は，突破成功の表示をする時間を調節してあげるためのものだ。WCはカウントするための変数名。WCが300未満で，かつキーボードからの文字入力が何もなかったら3110行を繰り返す。 WCが300以上になったり，入力があったらCLS文で画面を消去。再度，得点を表示して，ゲーム終了の表示になる。

```
2870 REM ET=2...victory
2880    COLOR C7,C4
2890    CLS
2900    LOCATE 21,0 :PRINT "POINT =";PNT;
2910    I$=INKEY$
2920                              '*----*----*----*----*----'
2930    LOCATE 7, 4 :PRINT "      │                 │ "
2940    LOCATE 7, 5 :PRINT " ┌────┴────┐ ─┬─  ┌──┼─┐"
2950    LOCATE 7, 6 :PRINT "│ │      │ │  │   │  │ │"
2960    LOCATE 7, 7 :PRINT "─┘    │   └─ ┘   │ ─┐ "
2970    LOCATE 7, 8 :PRINT "──────┼──────  │┌┐ │ ┐│ "
2980    LOCATE 7, 9 :PRINT "    ┌─┴─┐     ││││ │ ┼┘ "
2990    LOCATE 7,10 :PRINT "────┘   └───  ┘└┘ │ │└─"
3000                              '*----*----*----*----*----'
3010    LOCATE 7,12 :PRINT "      │  ＼          │  "
3020    LOCATE 7,13 :PRINT "  ┌───┼────── ─┬─  ─┼──┐"
3030    LOCATE 7,14 :PRINT "  ├─┐ └┐   ┌   │    │  │"
3040    LOCATE 7,15 :PRINT "  │ │  └┐┌─┘   │    │  │"
3050    LOCATE 7,16 :PRINT "  │ │   └┼┐    │    │  │"
3060    LOCATE 7,17 :PRINT "  │ │ ┌─┘└┐  ──┴── ┌┘  │"
3070    LOCATE 7,18 :PRINT "  │ ┘ ┘    └─     ┘  ─┘"
3080                              '*----*----*----*----*----'
3090    BEEP
3100    WC=0
3110    IF WC<300 AND INKEY$="" THEN WC=WC+1:GOTO 3110
3120    CLS
3130    LOCATE 21,0 :PRINT "POINT =";PNT;
3140   RETURN
3150 REM ET=4...time's up
3160   LOCATE 4,0 :COLOR C5 :PRINT "Time's up!"
3170   FOR I=1 TO 27 :BEEP 1 :FOR J=1 TO 15 :NEXT J :BEEP 0 :NEXT I
3180   RETURN
```



次の地雷源へ 地雷の数を増やす

ここでは，今までに触れることのできなかった場面を2つ設定することにしよう。1つは，第2，第3面目の地雷原。そしてもう1つは，地雷の爆発の場面だ。

2320～2410行は，第2，第3面目である。2330行では，地雷数を2つ増やしており，第2面で4＋2＝6個，第3面で6＋2＝8個となる。2340行では，現時点における残り時間をJに代入。2350～2390行で，残り時間を1ずつ減らし，残り時間を画面に表示。そして，GOSUB文で，得点のサブルーチンへ2度。1点×2＝2点得点する。2380行。2350, 2390行のFOR～NEXT文で，残り時間分くり返すから2点×残り時間数，つまり残り時間の2倍得点する。

2400行で，現在の地雷数が，8個（MBMBだね）より多いかどうかを判定している。8個より多ければ，第2，第3面を終了していることになる。従って，ゲーム終了だ。8個以下であれば，2410行で1090行へ行って，地雷原の設定をして，挑戦が始まる。これで，回を追うごとにスリリングになるゲーム化ができた。

```
2320    REM next screen
2330      NBMB=NBMB+2
2340      J=RT
2350      FOR I=J TO 1 STEP -1
2360       RT=RT-1
2370       LOCATE 10,0 :PRINT RT;
2380       GOSUB 3200:GOSUB 3200:'Point*2
2390      NEXT I
2400       IF NBMB>MBMB THEN ET=2:GOTO 2430:'Ending
2410      GOTO 1090:'................................Setting
```



爆発シーンも作ろう　放射状に伸びる爆発シーン

幸い乱たち３人は地雷を踏まなかった。しかし，地雷を踏んだら爆発するシーンも作らないと，ゲームとしては物足りないね。そこで，地雷を踏んだら爆発するシーンをどうやって作るか考えてみよう。

①地雷を踏んだら地雷があらわれる。―CIRCLE 文で何とかなりそうだ。

②次は爆発。これはハデにやろう。放射状にいろんな色の直線が伸びていくというのはどうだろう。―LINE文だね。

③爆発音も出さないと物足りない。―BEEPでいこう。

④地雷を踏んだのだから，もちろんこれでゲーム・オーバー*。――ENDの処理をするところへ飛んでいくことにしよう。

こんなものでどうだろう？

あとは手順よく①，②，③，④をプログラムしていくことだ。このためには，地雷の位置を覚えている変数も出番がありそうだね。その地雷の位置にCIRCLEを描けばいい。

放射線には乱数を使ってみよう。x方向，y方向とも，画面のドットの範囲内にするため，ちょっと細工も必要だ。x方向は０～639，y方向は０～199の数字がデタラメに出てくるようにするためにね。

さてこれらのことがイメージできたらリストを見てみよう。リストとその細かな解説は76ページに出ている。

＊　ゲーム・オーバー＝GAME OVERと書き，ゲーム終了という意味。ゲームの最後のシーンによく表示するメッセージだ。



GAME OVER ON GOSUB INPUT$

ゲームにも開始のタイトル，終了のメッセージというのは必要だ。最後は，この終了のメッセージ作りといこう。

終了するのは次の場合だったね。

①3人が3面を無事突破──→突破成功と表示してEND

②地雷を踏む──→爆発してEND

③時間切れ──→Time's　upと表示してEND

④自爆──→爆発してEND

この4つだ。これをプログラム化する手順を次に考えてみよう。

ポイントは4つの場合に応じて別々な処理をした後，終わらせることだね。このために使えそうな命令に，ON GOSUBというのがあることを知っておこう。

ON GOSUB：オン　ゴートゥサブルーチン*／on go to subroutineの省略形。on（～の時），go to　（～へ行く）だから，～の時はサブルーチンへ行くという意味。

《使い方》　ON　式　GOSUB　行番号①，行番号②，………………

式の値が1の時は①の行へ，2の時は②の行へということになる。

これを上手に使えば終わりの場合分けはなんとかなりそうだ。リストとその解説は76ページに掲載してあるから参照しよう。

＊　GOSUBはふつうゴーサブって読む人が多いので，正式には上の読み方なんだけど，オン　ゴーサブでも通じる。



PRINT文の秘密 PRINT〝*″, PRINT〝*″;

乱が何やらPRINT文の後に,(カンマ)や;(セミコロン)をくっつけているね。これは何だろう?

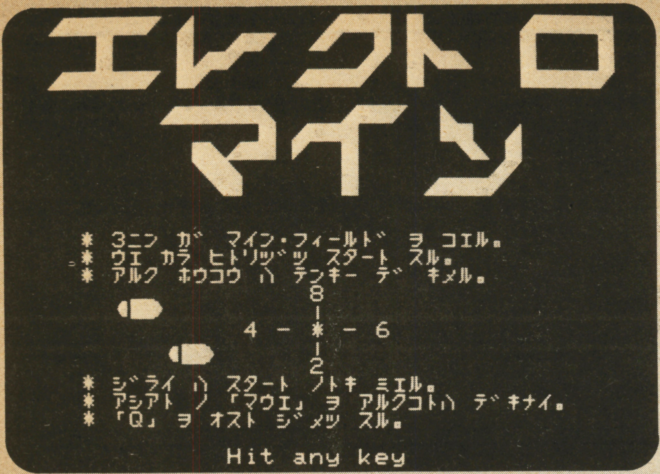
PRINT文というのは,改行する時にも使うのを覚えているかな。PRINTとだけ単独で使えば1行改行するわけだ。この応用みたいなもので,PRINT〝*″とやれば,*を表示して改行が行われるということを覚えておこう。

だったら,改行しないまま続けて出せるような命令もほしくなるのが人情というもの。そのためにあるのが〝,″と〝;″だ。この2つの違いは,〝,″は*を表示した後一定の間隔*をあけた次の位置に再び文字が表示できるようにする役目を持っているのに対し,〝;″は間隔をあけずにすぐ続けて表示する役目を持っている。これで乱のやってることがわかったかな?

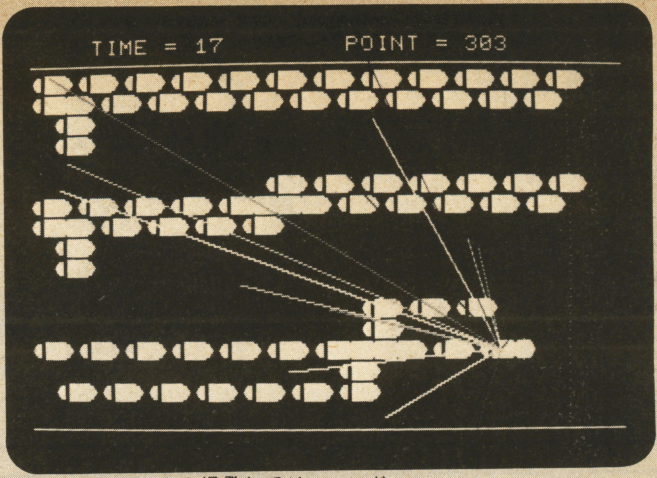
さて,この章もこれでEND。エレクトロマインゲームを少しずつブロックに分けながら説明してきたけれど,ここで,もう一度41ページのフローチャートを見てごらん。

このゲームがどのような構成になっているのかが改めてわかるだろう。小さな一つ一つのものは何とかプログラムできるはずだ。キミに必要なのは,それらの小さな仕事をどうつないで組み合わせるかということだ。これを常に念頭に置きながら,次ページからの全リストでここまでの整理をしよう。

* ,であく間隔は機種によって違うことがある。ふつうは8字あくものが多い。



▲タイトルの表示



▲地雷にふれたら爆発してジ・エンド

電子地雷エレクトロマイン

●遊び方

キミが、これから通り抜けようとする通路は、電子地雷路と化している。キミは埋められている地雷を爆発させることなく、ゴールに到達しなければならない。しかし、そう案ずることはない。一瞬ではあるが、埋められているはずの地雷が姿を現し、キミにその位置を教えてくれる。

地雷路は、えんえんと続いている。ゴールかと思えば，次の地雷路がキミを待ちかまえている。しかも，地雷の数を増やして。キミは，テンキーの2(↓)，4(←)，6(→)，8(↑)で，上下左右の方向へ自由に移動できる。一度歩いた上を再度歩くことはできない。また，〝Q〟を押すと自爆するので要注意！

移植とは？

この本のプログラムはすべて，FUJITSU MICRO 8（以下FM-8と略す）のF-BASIC（エフベーシック）で書かれている。ベーシックで書かれたものだから，ベーシックを使える機種であれば，どんな機種でもこのゲームを楽しめるはずだと思うね。ところが，ベーシックといっても，機種により少しずつ違いがあるんだ。

まず，ハードウエア上の違いだ。問題になるのは，ディスプレイを作っているドットの大きさ・数が，各機種によって異なることである。次に，ソフトウエア上の違いだ。ベーシックの種類によって，命令数や表現方法の違いなどがある。同じ働きをもつ命令でも，表現方法が異なっている場合があり，これを〝方言〟ともいう。

そこで，FM-8用に組んだこのゲームを，他機種でも使えるようにするために，〝移植〟という作業が必要になってくる。この本では，NECのPC-8001（以下[PC]，シャープのMZ-700([MZ])，東芝のパソピア([PAS])，日立のレベル3マークII([L3])に移植をした。各機種ごとの移植のポイントおよび変更する行の部分はFM-8以外のリストの後に続けてのせている。FM-8以外のマイコンを使っているキミは，その行をFM-8のものと書きかえて使おう。

(富士通) FM8 & FM7全リスト

```
10 REM *******************************
20 REM *                             *
30 REM *     ｴﾚｸﾄﾛ･ﾏｲﾝ ﾄｯﾊﾟ ｻｸｾﾝ      *
40 REM *                             *
50 REM *      for FM-8               *
60 REM *                             *
70 REM *      1983.3.12. by RAN      *
80 REM *                             *
90 REM *******************************
100 '
```

初期設定

```
110 REM ---------- Initialize ----------
120 'initialize FM-8
130   DEFINT A-Z
140   COLOR ,0
150   CLS
160   WIDTH 40,25                                        ── 画面のわくどり設定
170   CONSOLE 0,25,0,0
180 'set constant
190   C0=0 :C1=1 :C2=2 :C3=3 :C4=4 :C5=5 :C6=6 :C7=7     ── カラーコードの設定
200   NPSN=3
```

変　数　表　❶

BBD	地雷と地雷の距離	DX, DY	進む方向, 距離
BRY	地雷の設置の是非（論理式）	ET	エンディングのテーマ
C0〜C7	カラーコード	FALSE	偽　（論理式）
CB	地雷の色	FNRAN( )	整数の乱数（利用者が定義した関数）
CF( )	足跡の色		
CR	爆発を示す線の色		
CW	壁とゴールの色	FPR%( )	足跡のパターン
D	方向	I, J, K, L, M	FOR〜NEXT の変数
D$	方向, 自爆・ダミー・再プレーする, しない	I$	ダミー
		LX, LY	足跡パターンの寸法
		MBMB	地雷の数の最大値
DSZ	足跡のデータ	MX	横方向半歩の長さ

変　数　表　❷

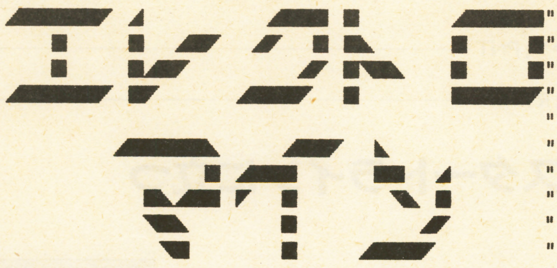
MY	縦方向１歩の長さ	ST	残り時間の初期値
N	データの個数	TC, WC	カウンタ
NBMB	各面での地雷数	TRUE	真　（論理式）
NPSN	人数	X, X1, X2, Y, Y1, Y2	座標
PBD	足跡から地雷までの歩数	XBMB( ), YBMB( )	地雷の座標
PNT	得点		
PSN	人の番号	XFT( ), YFT( )	足跡の座標
RB	地雷の半径		
RL	右足か左足か	XR, YR	爆発を示す線の終点の座標
RT	残り時間		
SIZE	FPR%( ) の添字の最大値	XT, YT	次に足跡が置かれる座標

```
210   NBMB=4
220   MBMB=NBMB+4
230   RB=12
240   MX=25 :MY=10
250   LX=MX*2 :LY=MY*2
260   DSZ=(((LX+1)*(LY+1)+7)/8+2-1)/2
270   ST=90
280   FALSE=0 :TRUE=NOT FALSE
290 'dimension
300   DIM FPR%(DSZ)
310   DIM CF(NPSN)
320   DIM XFT(NPSN,2),YFT(NPSN,2)
330   DIM XBMB(MBMB),YBMB(MBMB)
340 'color
350   CF(0)=C6
360   CF(1)=C2
370   CF(2)=C6
380   CF(3)=C1
390   CW=C7
400   CB=C3
410 'randomize
420   RANDOMIZE(TIME/3)
430 'function
440   DEF FNRAN(SIZE)=INT(RND(1)*SIZE+1)
```

220: 地雷の最大数
300: 足跡のパターンをしまっておく配列
330: 地雷の x 座標と y 座標の配列
390: 壁の色
400: 地雷の色

タイトル表示

```
450 '
460 REM ---------- Title ----------
470 'how to play
480   COLOR C3
490         '*-----*-----*-----*-----*-----*-----*-----*-'
500   PRINT "                                        "
510   PRINT "                                        "
520   PRINT "                                        "
530   PRINT "                                        "
540   PRINT "                                        "
550   PRINT "                                        "
560   PRINT "                                        "
570   PRINT "                                        "
580   PRINT "                                        "
590   PRINT "                                        "
600         '*-----*-----*-----*-----*-----*-----*-----*-'
```



```
610   COLOR C7
620   PRINT :PRINT
630   PRINT "     * 3ニン ガ マイン・フィールド ヲ コエル。"
640   PRINT "     * ウエ カラ ヒトリヅツ スタート スル。"
650   PRINT "     * アルク ホウコウ ハ テンキー デ キメル。"
660   PRINT TAB(14);"     8"
670   PRINT TAB(14);"     ¦"
680   PRINT TAB(14);"4 - * - 6"
690   PRINT TAB(14);"     ¦"
700   PRINT TAB(14);"     2"
710   PRINT "     * ジライ ハ スタート ノトキ ミエル。"
720   PRINT "     * アシアト ノ 「マウエ」 ヲ アルクコトハ デキナイ。"
730   PRINT "     * 「Q」 ヲ オスト ジメツ スル。"
```

490: タイトルの表示
630: 遊び方の説明表示

3人の足跡作成

```
740 'making footprint
750   X=100 :Y=120
760   FOR L=1 TO 2
770    READ N                                                線をひくためのデータの個数を読む
780    READ X1,Y1                                            線をひくためのデータを読む
790    FOR M=1 TO N-1
800     READ X2,Y2                                           線をひくためのデータを読む
810     LINE (X1+X,Y1+Y)-(X2+X,Y2+Y),PSET,CF(0)
820     X1=X2 :Y1=Y2
830    NEXT M
840    READ X1,Y1
850    PAINT (X1+X,Y1+Y),CF(0)                               ペイントするためのデータを読む
860   NEXT L
870   BEEP 1 :BEEP 0
880   GET@ (X,Y)-(X+LX,Y+LY),FPR%,G,CF(0)                    足跡のパターンを読み込む
890   FOR I=1 TO 140 :NEXT I
900   BEEP 1 :BEEP 0
910   PUT@ (X+LX,Y+LY)-(X+LX*2,Y+LY*2),FPR%,PSET,CF(0)       2歩目の足跡を表示
920 'footprint data
930  'number of data (1)
940   DATA 8
950  'line data (1)
960   DATA 11,7, 8,7, 5,8, 3,10, 5,12, 8,13, 11,13, 11,7
970  'paint data (1)
980   DATA 10,8
990  'number of data (2)
1000   DATA 8
1010  'line data (2)
1020   DATA 16,6, 16,14, 36,14, 41,13, 46,10, 41,7, 36,6, 16,6
1030  'paint data (2)
1040   DATA 17,7
1050 'wait for inkey
1060  LOCATE 13,24 :COLOR C4 :PRINT "Hit any Key";
1070  IF INKEY$="" GOTO 1070
1080 '
```

地雷路の壁を作成

```
1090 REM ---------- Setting ----------
1100 REM Frame
1110 'wall
1120   COLOR ,C0
1130   CLS
1140   LINE (0,12)-(639,12),PSET,CW                          上の壁の線をひく
1150   LINE (0,188)-(639,188),PSET,CW                        下の壁の線をひく
1160   FOR Y=13 TO 188 STEP 4
1170    PSET (23*MX+1,Y,CW)
1180   NEXT Y
1190 'point
1200   COLOR C7
1210   LOCATE 21,0 :PRINT "POINT =";PNT                      得点表示
1220   RT=ST
1230   LOCATE 4,0 :PRINT "TIME =";RT                         残り時間表示
```

3人がスタートラインに立つ

```
1240 REM Footprint
1250 'initial condition
1260   PSN=1 :RL=2                                           PSN = 1は1人目, RL = 2は右足
1270   XFT(1,1)=1 :YFT(1,1)=3
1280   XFT(1,2)=2 :YFT(1,2)=5
1290   XFT(2,1)=1 :YFT(2,1)=9
1300   XFT(2,2)=2 :YFT(2,2)=11
1310   XFT(3,1)=1 :YFT(3,1)=15
1320   XFT(3,2)=2 :YFT(3,2)=17
1330 'put footprint
1340   FOR I=1 TO 3
1350    FOR J=1 TO 2
1360     X=XFT(I,J)*MX :Y=YFT(I,J)*MY
1370     PUT@ (X-MX,Y-MY)-(X+MX,Y+MY),FPR%,PSET,CF(I)
1380    NEXT J
1390   NEXT I
```

地雷が見えた！

```
1400 REM Mine
1410 'lay mines for free space at random
1420   FOR I=1 TO NBMB
1430    REM serch
1440     XBMB(I)=FNRAN(19)+3
1450     YBMB(I)=FNRAN(17)+1
1460     BRY=TRUE
1470     J=1
1480     IF BRY AND J<I THEN 1490 ELSE 1530
1490      BBD=ABS(XBMB(I)-XBMB(J))+ABS(YBMB(I)-YBMB(J))
1500      IF BBD<=5 THEN BRY=FALSE
1510      J=J+1
1520     GOTO 1480
1530    IF NOT BRY GOTO 1430:'serch
1540    FOR J=1 TO 10 :BEEP 1 :BEEP 0 :NEXT J
1550    CIRCLE (XBMB(I)*MX,YBMB(I)*MY),RB,CB,,,,F,PSET
1560   NEXT I
1570 'hide mines
1580   FOR I=1 TO NBMB
1590    CIRCLE (XBMB(I)*MX,YBMB(I)*MY),RB,CB,,,,F,PRESET
1600   NEXT I
1610 '
```

1440: 地雷のx座標
1450: 地雷のy座標
1490: 地雷間の距離を測る
1550: 地雷の表示
1590: 地雷の消去

スタートの合図

```
1620 REM ---------- Start ----------
1630 'start!
1640   D$=INKEY$
1650   LOCATE 14,24 :COLOR CF(PSN) :PRINT "S T A R T";
1660   FOR I=1 TO 180 :BEEP 1 :BEEP 0 :NEXT I
1670   LOCATE 14,24 :PRINT "         ";
1680   COLOR C7
1690 '
```

1650: “スタート”の表示

歩く方向は？

```
1700 REM ---------- Walk ----------
1710 REM Input
1720 'get character  'add counter
1730   TC=TC+1 :IF TC>20 THEN TC=0:GOSUB 3270:'Time
1740   D$=INKEY$
1750   IF D$<>"2" AND D$<>"4" AND D$<>"6" AND D$<>"8" AND D$<>"Q" GOTO 1730
1760   TC=TC+20
1770 'set direction and right-left
1780   IF D$="Q" THEN ET=3:GOTO 2430:'Ending
1790   D=VAL(D$)
1800   DX=0 :DY=0
1810   ON D¥2 GOTO 1820,1860,1900,1940
1820   REM D=2...down
1830     DY=1
1840     IF YFT(PSN,1)<YFT(PSN,2)-1 THEN RL=1 ELSE RL=2
1850     GOTO 1980
1860   REM D=4...left
1870     DX=-2
1880     IF XFT(PSN,1)<XFT(PSN,2) THEN RL=2 ELSE RL=1
1890     GOTO 1980
1900   REM D=6...right
1910     DX=2
1920     IF XFT(PSN,1)<XFT(PSN,2) THEN RL=1 ELSE RL=2
1930     GOTO 1980
1940   REM D=8...up
1950     DY=-1
1960     IF YFT(PSN,1)<YFT(PSN,2)-1 THEN RL=2 ELSE RL=1
1970     GOTO 1980
```

1730: TC＞20になると時間が1減る
1840: 左右どちらの足を動かすかの判定

次の歩きを進める

```
1980 REM Step
1990 'step?
2000   XT=XFT(PSN,RL)+DX :YT=YFT(PSN,RL)+DY
2010    IF YT<=1 OR YT=>19 OR XT<=0 GOTO 1700:'......Walk
2020   X=XT*MX :Y=YT*MY
2030    IF POINT(X,Y) GOTO 1700:'....................Walk
```

2010: 上，下，左の範囲を越えていないか
2030: すでに足跡があるかの判定

```
2040 'step
2050   XFT(PSN,RL)=XT :YFT(PSN,RL)=YT
2060   PUT@ (X-MX,Y-MY)-(X+MX,Y+MY),FPR%,PSET,CF(PSN)
2070   GOSUB 3200:'Point
2080 'person-bomb distance
2090   FOR I=1 TO NBMB
2100     PBD=ABS((XBMB(I)-XFT(PSN,RL)))¥2+ABS(YBMB(I)-YFT(PSN,RL))
2110     IF PBD>2 GOTO 2150
2120       IF PBD=0 THEN ET=1:GOTO 2430:'Ending
2130       IF PBD=1 THEN GOSUB 3200:GOSUB 3200:'Point*2
2140       IF PBD=2 THEN GOSUB 3200:'Point
2150   NEXT I
```

2050: 位置を配列に入れる
2100: 足跡と地雷の距離

ゴールに到着したか

```
2160 REM Reach
2170 'reach?
2180   IF XFT(PSN,RL)<23 GOTO 1700:'.................Walk
2190 'next person or next screen
2200   FOR I=1 TO 20 :GOSUB 3200 :NEXT I :'Point*20
2210   PSN=PSN+1
2220   IF PSN>NPSN GOTO 2320
2230   REM next person
2240     FOR I=1 TO NBMB
2250      FOR J=1 TO 10 :BEEP 1 :BEEP 0 :NEXT J
2260      CIRCLE (XBMB(I)*MX,YBMB(I)*MY),RB,CB,,,,F,PSET
2270     NEXT I
2280     FOR I=1 TO NBMB
2290      CIRCLE (XBMB(I)*MX,YBMB(I)*MY),RB,CB,,,,F,PRESET
2300     NEXT I
2310     GOTO 1620:'...............................Start
2320   REM next screen
2330     NBMB=NBMB+2
2340     J=RT
2350     FOR I=J TO 1 STEP -1
2360      RT=RT-1
2370      LOCATE 10,0 :PRINT RT;
2380      GOSUB 3200:GOSUB 3200:'Point*2
2390     NEXT I
2400      IF NBMB>MBMB THEN ET=2:GOTO 2430:'Ending
2410     GOTO 1090:'...............................Setting
2420 '
```

2220: 次の人間がスタートできるかの判定
2260: 次の人間がスタートする前に地雷を表示
2330: 地雷を2個増やす
2380: 残り時間の2倍の数を得点

ゲームの終了

```
2430 REM ---------- Ending ----------
2440 REM ending main routine
2450   ON ET GOSUB 2700,2870,2720,3150
2460   COLOR C3
2470   LOCATE 3, 7 :PRINT "#########    #        #    ###"
2480   LOCATE 3, 8 :PRINT "#            ##       #    #  ##"
2490   LOCATE 3, 9 :PRINT "#            # #      #    #    #"
2500   LOCATE 3,10 :PRINT "#            #  #     #    #     #"
2510   LOCATE 3,11 :PRINT "########     #   #    #    #     #"
2520   LOCATE 3,12 :PRINT "#            #    #   #    #     #"
2530   LOCATE 3,13 :PRINT "#            #     #  #    #     #"
2540   LOCATE 3,14 :PRINT "#            #      # #    #    #"
2550   LOCATE 3,15 :PRINT "#            #       ##    #  ##"
2560   LOCATE 3,16 :PRINT "#########    #        #    ###"
2570   D$=INKEY$
2580   BEEP
2590   LOCATE 12,24 :COLOR C7 :PRINT "Replay? (Y or N) ";
2600   D$=INPUT$(1)
2610    PRINT D$;
2620    COLOR ,C0
2630    IF D$="N" OR D$="n" GOTO 2670
2640    IF D$="Y" OR D$="y" THEN 2650 ELSE 2580
2650    NBMB=4 :PNT=0
2660    GOTO 1090:'...............................Setting
2670   CLS
2680   BEEP
2690  END
```

2470: ゲーム終了の表示
2590: 再度,ゲームをしますか？

地雷の爆発

```
2700 REM ET=1...strike a mine
2710   CIRCLE (XBMB(I)*MX,YBMB(I)*MY),RB,CB,,,,F
```

2710: 踏んだときの地雷表示

```
2720 REM ET=3...self-destruction
2730   BEEP
2740   FOR I=1 TO 20
2750    XR=FNRAN(640)-1
2760    YR=FNRAN(200)-1
2770    CR=FNRAN(7)
2780    LINE (XFT(PSN,RL)*MX,YFT(PSN,RL)*MY)-(XR,YR),PSET,CR
2790    FOR J=1 TO 2
2800     BEEP 1
2810     FOR K=1 TO FNRAN(4) :NEXT K
2820     BEEP 0
2830     IF ET=3 THEN GOSUB 3200:'Point
2840    NEXT J
2850   NEXT I
2860  RETURN
```

爆発の不規則な音

突破成功！

```
2870 REM ET=2...victory
2880   COLOR C7,C4
2890   CLS
2900   LOCATE 21,0 :PRINT "POINT =";PNT;
2910   I$=INKEY$
2920                     '*----*----*----*----*----'
2930   LOCATE 7, 4 :PRINT "      |               |  "
2940   LOCATE 7, 5 :PRINT " ┌────┴─────┐ ──┬── ┌──┼─┐"
2950   LOCATE 7, 6 :PRINT "| |        | |  |   |  | |"
2960   LOCATE 7, 7 :PRINT "──┘   |    └──  ┌┘  | ──┐ "
2970   LOCATE 7, 8 :PRINT "──────┼──────── |┌─┐ | ─┐| "
2980   LOCATE 7, 9 :PRINT "    ┌─┴─┐       ||  | | ┌┼┘ "
2990   LOCATE 7,10 :PRINT "────┘    └──── ┘└──┘ | |└──"
3000                     '*----*----*----*----*----'
3010   LOCATE 7,12 :PRINT "      |   \           |  "
3020   LOCATE 7,13 :PRINT "  ┌───┼────── ──┬── ──┼──┐"
3030   LOCATE 7,14 :PRINT "  ├──┐ └┐   ┌    |    |  |"
3040   LOCATE 7,15 :PRINT "  |  |  └┐ ┌──┘  |    |  |"
3050   LOCATE 7,16 :PRINT "  |  |   └┼┐     |    |  |"
3060   LOCATE 7,17 :PRINT "  |  | ┌──┘└┐  ──┴── ┌─┘  |"
3070   LOCATE 7,18 :PRINT "  | ┘ ┘    └─       ┘   ─┘"
3080                     '*----*----*----*----*----'
3090   BEEP
3100   WC=0
3110   IF WC<300 AND INKEY$="" THEN WC=WC+1:GOTO 3110
3120   CLS
3130   LOCATE 21,0 :PRINT "POINT =";PNT;
3140  RETURN
```

"突破成功"の表示

ゲームの得点を表示

時間切れのサブルーチン

```
3150 REM ET=4...time's up
3160   LOCATE 4,0 :COLOR C5 :PRINT "Time's up!"
3170   FOR I=1 TO 27 :BEEP 1 :FOR J=1 TO 15 :NEXT J :BEEP 0 :NEXT I
3180  RETURN
3190 '
```

時間切れを知らせる

得点のサブルーチン

```
3200 REM ---------- Point ----------
3210 'point increment
3220   BEEP 1 :BEEP 0
3230   PNT=PNT+1
3240   LOCATE 28,0 :PRINT PNT;
3250  RETURN
3260 '
```

残り時間のサブルーチン

```
3270 REM ---------- Time ----------
3280 'time decrement
3290   RT=RT-1
3300   LOCATE 10,0 :PRINT RT;
3310   IF RT>=10 GOTO 3340
3320    IF RT=0 THEN ET=4:RETURN 2430:'Ending
3330    IF RT<=9 THEN BEEP
3340  RETURN
```

残り時間＝0でゲームは終了

(NEC) PC8001への移植のPoint

CONSOLE w, x, y, z　*w*, *x*, *y*, *z*をそれぞれ第1パラメータ，第2パラメータ，第3パラメータ，第4パラメータという。第4パラメータのカラーモードをFM-8は0で[PC]は1で設定する。

画面構成(ドット数)　FM-8は640×200，[PC]は160×100(WIDTH　80のとき) か，80×100(WIDTH　40のとき) 換算の必要がある。このゲームでは，80×100に換算。

COLOR x, y　[PC]は第2パラメータを持たないので背景色の指定はできない。

CIRCLE　[PC]にはない。BF(またはB) でLINEを使う。

PAINT　[PC]にはないので，PSETやLINE文を用いる。

RANDOMIZE　[PC]にはないので，乱数の系列は変えない。

RETURN 〈行番号〉　行番号の指定はできないので，GOTOを使う。

INKEY$　オートリピートがきかないので，キーを押しっぱなしにしていても入力できるようにINPを使う。

CLS　[PC]にはないのでPRINT CHR$(12)を使う。

GET@, PUT@　PUT@での色指定はできないので，直前にCOLOR文を置く。

```
140   COLOR ,0,1
150   PRINT CHR$(12);

170   CONSOLE 0,25,0,1

240   MX=3 :MY=5

260   DSZ=(LX¥2)*(LY¥4)

410 ''
420 ''

750   X=12 :Y=60

840    ''
850    ''

880   GET@ (X,Y)-(X+LX,Y+LY),FPR%,G
890   FOR I=1 TO 140 :NEXT I

910   COLOR CF(0) :PUT@ (X+LX,Y+LY-1)-(X+LX*2,Y+LY*2-1),FPR%,PSET

940   DATA 2

960   DATA 1,4,1,6
970  ''
980   ''

1000   DATA 4

1020   DATA 3,3, 3,7, 4,7, 4,3
1030  ''
1040   ''

1120   '
1130   PRINT CHR$(12);
1140   LINE (0,6)-(79,6),PSET,CW
1150   LINE (0,94)-(79,94).PSET.CW
1160    FOR Y=7 TO 94 STEP 4
1370     COLOR CF(I) :PUT@ (X-MX,Y-MY)-(X+MX,Y+MY),FPR%,PSET

1550    LINE (XBMB(I)*MX-1,YBMB(I)*MY-1)-(XBMB(I)*MX+1,YBMB(I)
        *MY+1),PSET,CB,BF

1590    LINE (XBMB(I)*MX-1,YBMB(I)*MY-1)-(XBMB(I)*MX+1,YBMB(I)
        *MY+1),PRESET,CB,BF

1740   D=0 :D$=INKEY$ :IF D$="Q" THEN ET=3 :GOTO 2430 :'Ending
```

```
1750    IF INP(0)=251 THEN D=2
1751    IF INP(0)=239 THEN D=4
1752    IF INP(0)=191 THEN D=6
1753    IF INP(1)=254 THEN D=8

1760   IF D<>2 AND D<>4 AND D<>6 AND D<>8 THEN 1730 ELSE TC=TC+20

1780   ''
1790   ''

2060   COLOR CF(PSN) :PUT@ (X-MX,Y-MY)-(X+MX,Y+MY),FPR%,OR

2260    LINE (XBMB(I)*MX-1,YBMB(I)*MY-1)-(XBMB(I)*MX+1,YBMB(I)
        *MY+1),PSET,CB,BF

2290    LINE (XBMB(I)*MX-1,YBMB(I)*MY-1)-(XBMB(I)*MX+1,YBMB(I)
        *MY+1),PRESET,CB,B
F

2670   PRINT CHR$(12);

2710   LINE (XBMB(I)*MX-1,YBMB(I)*MY-1)-(XBMB(I)*MX-1,YBMB(I)
       *MY+1),PSET,CB

2750    XR=FNRAN(80)-1
2760    YR=FNRAN(100)-1

2890   PRINT CHR$(12);

3120   PRINT CHR$(12);

3310   IF RT>=10 GOTO 3340
```

(日立) LEVEL3への移植のPoint

CONSOLE w, x, y, z　[L3]には第4パラメータはない。いつでもカラーモード。

画面構成(ドット数)　640×200に換算が必要。

CIRCLE　[L3]にはない。三角関数のsin, cosとFOR～NEXTで座標を決め，PSETを使って円を描く。

PAINT　[L3]のPAINTは，ドット以外を境界とみなさない。

GET@, PUT@　[L3]にはない。LOCATを使う。

BEEP 1, BEEP 0　[L3]は，1，0の指定ができない。一定時間だけ音を出すBEEPだけである。

WIDTH x, y　第2パラメータのyはない。行数は25で固定。

モード　SCREENの第1パラメータでドット数のこまかいハイレゾリューションモードを，第3パラメータでキャラクタを表示するノンインターレースモードを指定する。

```
130   DEFINT A-Y

150   CLS :SCREEN 1,1,1
160   WIDTH 40
170   CONSOLE 0,25,0

260   ''

300   ''

750   X=100:Y=120:C=0:GOSUB 760 :X=100+LX:Y=120+LY:C=0:GOSUB 760
      :GOTO 1050
```

```
760   RESTORE 920 :FOR L=1 TO 2

810    LINE (X1+X,Y1+Y)-(X2+X,Y2+Y),PSET,CF(C)

850   PAINT (X1+X,Y1+Y),CF(C)

870   BEEP :RETURN
880   ''
890   ''
900   ''
910   ''
```

```
980 DATA 9,8
1040 DATA 18,7
1370     X=X-MX:Y=Y-MY:C=I:GOSUB 760
1550    FOR Z=0 TO 6.29 STEP .15 :PSET(XBMB(I)*MX+SIN(Z)*RB,
        YBMB(I)*MY+COS(Z)*RB /3,CB) :NEXT Z
1590    FOR Z=0 TO 6.29 STEP .15 :PRESET(XBMB(I)*MX+SIN(Z)*RB,
        YBMB(I)*MY+COS(Z)* RB/3,CB) :NEXT Z
1640   FOR I=1 TO 31 :D$=INKEY$ :NEXT I
1660   BEEP :BEEP
2060   X=X-MX:Y=Y-MY:C=PSN:GOSUB 760
2250      BEEP
2260      FOR Z=0 TO 6.29 STEP .15 :PSET(XBMB(I)*MX+SIN(Z)*RB,
       YBMB(I)*MY+COS(Z)* RB/3,CB) :NEXT Z
2290      FOR Z=0 TO 6.29 STEP .15 :PRESET(XBMB(I)*MX+SIN(Z)*RB,
       YBMB(I)*MY+COS(Z)*RB/3,CB) :NEXT Z
2570   FOR I=1 TO 31 :D$=INKEY$ :NEXT I
2710   FOR Z=0 TO 6.29 STEP .15 :PSET(XBMB(I)*MX+SIN(Z)*RB,
       YBMB(I)*MY+COS(Z)*RB/3,CB) :NEXT Z
```

```
2780    LINE (XFT(PSN,RL)*MX,YFT(PSN,RL)*MY)-(XR,YR),PSET,CR
2790    ''
2800     BEEP
2810     FOR K=1 TO FNRAN(8) :NEXT K
2820     ''
2840    ''
3170   FOR I=1 TO 7 :BEEP :FOR J=1 TO 15 :NEXT J :NEXT I
3220   BEEP
```

(シャープ)MZ700への移植のPoint

CONSOLE w, x, y, z MZでは，第3パラメータで横スクロールの開始，第4パラメータZでスクロール桁数を設定する。

画面構成(ドット数) 80×50に換算が必要。

CIRCLE MZにはない。FOR~NEXTを使ってPSETで表示する。

PAINT MZにはない。PSETで表示するときに，カラーコードを設定。

GET@, PUT@ MZにはない。PSETを使う。

LINE MZにはない。FOR~NEXT と PSETを使う。

WIDTH x, y : RANDOMIZE MZにはない。

FOR~NEXT MZでは，FOR~NEXT のループからの飛び出しはできない。

INKEY$ MZでは，キー入力をするときに，小文字の入力はできない。

INPUT$ MZでは，キー入力するときに，小文字の入力はできない。

カーソルコントロールキー MZには，⬅➡などのキャラクタがあり，テンキーのかわりに，カーソルコントロールキーを使った。

```
120 'INITIALIZE MZ-700
160  ''
170  CONSOLE 0,25,0,40
230  ''
240  MX=3 :MY=2
260  ''
300   ''
410 ''
420   ''
650   PRINT "     * ｱﾙｸ ﾎｳｺｳ ﾊ ｶｰｿﾙｺﾝﾄﾛｰﾙ
ｷｰ ﾃﾞ ｷﾒﾙ｡"
660   PRINT
670   PRINT TAB(16);"  ↑"
680   PRINT TAB(16);"← * →"
690   PRINT TAB(16);"  ↓"
700   PRINT
720   ''
750   X=12:Y=32:C=CF(0):GOSUB 760 :X=12+
LX:Y=32+LY:C=CF(0):GOSUB 760 :GOTO 1050
760   RESTORE 920 :FOR L=1 TO 2
780    ''
810     PSET (X+X2,Y+Y2,C)
820     ''
840    ''
850    ''
870   BEEP 1 :BEEP 0 :RETURN
880   ''
890   ''
900   ''
910   ''
940   DATA 2
950  'PSET DATA (1)
960   DATA 0,0,0,1
970  ''
980   ''
1000   DATA 2
1010  'PSET DATA (2)
1020   DATA 2,0, 2,1
1030  ''
1040   ''
1140   FOR X=0 TO 79 :PSET(X,2,CW) :NEXT
 X
1150   FOR X=0 TO 79 :PSET(X,45,CW) :NEX
T X
1160   FOR Y=4 TO 44 STEP 4
1270   XFT(1,1)=1 :YFT(1,1)=4
1280   XFT(1,2)=2 :YFT(1,2)=6
1290   XFT(2,1)=1 :YFT(2,1)=10
1300   XFT(2,2)=2 :YFT(2,2)=12
```

```
1310   XFT(3,1)=1 :YFT(3,1)=16
1320   XFT(3,2)=2 :YFT(3,2)=18
1370     C=CF(I):GOSUB 760
1450     YBMB(I)=FNRAN(20)+1
1550    X=XBMB(I)*MX :Y=YBMB(I)*MY :GOSU
B 3350:'MINE(LAY)
1590    X=XBMB(I)*MX :Y=YBMB(I)*MY :GOSU
B 3410:'MINE(HIDE)
1740   D$=INKEY$(0)
1750   IF D$<>"↓" AND D$<>"←" AND D$<>"→
" AND D$<>"↑" AND D$<>"Q" GOTO 1730
1790   IF D$="↓" THEN D=2:GOTO 1800
1791   IF D$="←" THEN D=4:GOTO 1800
1792   IF D$="→" THEN D=6:GOTO 1800
1793   IF D$="↑" THEN D=8
2010    IF YT<=1 OR YT=>22 OR XT<=0 GOTO
 1700:'......WALK
2060   C=CF(PSN):GOSUB 760
2090   I=1
2150   I=I+1 :IF I<=NBMB GOTO 2100
2190 'NEXT PERSON OR NEXT SCREEN
2260       X=XBMB(I)*MX:Y=YBMB(I)*MY:GOSU
B 3350:'MINE(LAY)
2290       X=XBMB(I)*MX:Y=YBMB(I)*MY:GOSU
B 3410:'MINE(HIDE)
2630     IF D$="N" GOTO 2670
2640     IF D$="Y" THEN 2650 ELSE 2580
2710   X=XBMB(I)*MX:Y=YBMB(I)*MY:GOSUB 3
350:'MINE(LAY)
2750     XR=XFT(PSN,RL)*MX+(FNRAN(10)-5):
IF XR<0 OR XR>79  GOTO 2750
2760     YR=YFT(PSN,RL)*MY+(FNRAN(5)-2):I
F YR<0 OR YR>49 GOTO 2760
2780    PSET (XR,YR,CR)
2810      FOR K=1 TO FNRAN(400) :NEXT K
3350 REM ---------- MINE (LAY) ----------
3360  FOR XX=-1 TO 3
3370   PSET(X+XX,Y,CB)
3380   PSET(X+XX,Y+1,CB)
3390  NEXT XX
3400 RETURN
3410 REM ---------- MINE (HIDE) ---------
--
3420  FOR XX=-1 TO 3
3430   PRESET(X+XX,Y)
3440   PRESET(X+XX,Y+1)
3450  NEXT XX
3460 RETURN
```

(東芝)PASOPIAへの移植のポイント

CONSOLE w, x, y, z [PAS] にはない。

COLOR [PAS] では，1行単位で実行される。1行の中で色を変えたいときは，カラー・アトリビュート・キャラクタをPRINT文で表示する。表示した位置から右側が指定した色になる。

GET@, PUT@ [PAS] では，ディスクを用いて走らせるとき以外は使えない。このゲームではサブルーチン化して，必要なときにそこに飛ばしている。

BEEP 1, BEEP 0 [PAS] では，これにあたるのが，SOUND *x*, *y*であり，*x*は音の高さ，*y*は音の長さ。

モード ドットのこまかいファイングラフィックモードをSCREENにて指定。

WIDTH x, y [PAS] には，第2パラメータはない。25行で固定。第1パラメータは，1～80まで指定できるが，このゲームでは，80文字を設定している。PRINT文での表示のずれはTABやLOCATEで修正する。

LINE [PAS] では，キャラクタで線をひくことはできないので，線をひくときに必要なPSETは不要になる。

```
160   WIDTH 80
170   SCREEN 2

230   RB=11

260   ''

300   ''

350   ''

500   PRINT TAB(16);"                                  "
510   PRINT TAB(16);"                                  "
520   PRINT TAB(16);"                                  "
530   PRINT TAB(16);"                                  "
540   PRINT TAB(16);"                                  "
550   PRINT TAB(16);"                                  "
560   PRINT TAB(16);"                                  "
570   PRINT TAB(16);"                                  "
580   PRINT TAB(16);"                                  "
590   PRINT TAB(16);"                                  "

640   PRINT TAB(16);"     * ウエ カラ ヒトリヅ゛ツ スタート スル。"
650   PRINT TAB(16);"     * アルク ホウコウ ハ テンキー デ゛ キメル。"
660   PRINT TAB(30);CHR$(255);"     8"
670   PRINT TAB(30);CHR$(255);"     ¦"
680   PRINT TAB(30);CHR$(255);"4 - * - 6"
690   PRINT TAB(30);CHR$(255);"     ¦"
700   PRINT TAB(30);CHR$(255);"     2"
710   PRINT TAB(16);"     * シ゛ライ ハ スタート ノトキ ミエル。"
720   PRINT TAB(16);"     * アシアト ノ 「マウエ」 ヲ アルクコトハ テ゛キナイ。"
730   PRINT TAB(16);"     * 「Q」 ヲ オスト シ゛メツ スル。"

750 I=1:X=100:Y=120:GOSUB 760 :I=2:X=140:Y=140:GOSUB 760
    :I=3:X=180:Y=120:GOSUB
760 :GOTO 1050
760   RESTORE 920 :FOR L=1 TO 2
810 LINE (X1+X,Y1+Y)-(X2+X,Y2+Y),CF(I)
850 PAINT (X1+X,Y1+Y),CF(I)
870   SOUND 45,6 :RETURN
880   ''
890   ''
900   ''
910   ''

1060  LOCATE 30,24 :COLOR C7 :PRINT "Hit any key";

1140   LINE (0,12)-(639,12),CW
1150   LINE (0,188)-(639,188),CW

1170    PSET (23*MX+1,Y),CW

1210   LOCATE 51,0 :PRINT "POINT =";PNT
1230   LOCATE 14,0 :PRINT "TIME =";RT

1370     X=X-MX:Y=Y-MY:GOSUB 760

1540    SOUND 60,10

1550    CIRCLE (XBMB(I)*MX,YBMB(I)*MY),RB,CB

1590    CIRCLE (XBMB(I)*MX,YBMB(I)*MY),RB,C0

1640   FOR I=1 TO 31 :D$=INKEY$ :NEXT I

1660   SOUND 55,30 :FOR I=1 TO 600:NEXT I

2060   I=PSN:X=X-MX:Y=Y-MY:GOSUB 760

2250      SOUND 60,10
2260      CIRCLE (XBMB(I)*MX,YBMB(I)*MY),RB,CB

2290      CIRCLE (XBMB(I)*MX,YBMB(I)*MY),RB,C0

2370      LOCATE 20,0 :PRINT RT;

2470   LOCATE 16, 7 :PRINT CHR$(251);"######### #        # ###"
2480   LOCATE 16, 8 :PRINT CHR$(251);"#         ##       # #  ##"
2490   LOCATE 16, 9 :PRINT CHR$(251);"#         # #      # #    #"
2500   LOCATE 16,10 :PRINT CHR$(251);"#         #  #     # #     #"
2510   LOCATE 16,11 :PRINT CHR$(251);"######## #   #    # #     #"
2520   LOCATE 16,12 :PRINT CHR$(251);"#         #    #   # #     #"
2530   LOCATE 16,13 :PRINT CHR$(251);"#         #     #  # #     #"
2540   LOCATE 16,14 :PRINT CHR$(251);"#         #      # # #    #"
2550   LOCATE 16,15 :PRINT CHR$(251);"#         #       ### #  ##"
2560   LOCATE 16,16 :PRINT CHR$(251);"######### #        # ###"
2570   FOR I=1 TO 31 :D$=INKEY$ :NEXT I
2580   SOUND 65,10
2590   LOCATE 30,24 :COLOR C7 :PRINT "Replay? (Y or N) ";

2670   SCREEN 0
2680   SOUND 65,10

2710   CIRCLE (XBMB(I)*MX,YBMB(I)*MY),RB,CB

2730   SOUND 70,20

2780    LINE (XFT(PSN,RL)*MX,YFT(PSN,RL)*MY)-(XR,YR),CR

2800     ''
2810     SOUND FNRAN(82),FNRAN(20)
2820     ''

2880   COLOR C1,C4
2900   LOCATE 51,0 :PRINT "POINT =";PNT;

2930   LOCATE 24, 4 :PRINT "                                 "
2940   LOCATE 24, 5 :PRINT "                                 "
2950   LOCATE 24, 6 :PRINT "                                 "
2960   LOCATE 24, 7 :PRINT "                                 "
2970   LOCATE 24, 8 :PRINT "                                 "
2980   LOCATE 24, 9 :PRINT "                                 "
2990   LOCATE 24,10 :PRINT "                                 "
3000                       '*----*----*----*----*----'
3010   LOCATE 24,12 :PRINT "                                 "
3020   LOCATE 24,13 :PRINT "                                 "
3030   LOCATE 24,14 :PRINT "                                 "
3040   LOCATE 24,15 :PRINT "                                 "
3050   LOCATE 24,16 :PRINT "                                 "
3060   LOCATE 24,17 :PRINT "                                 "
3070   LOCATE 24,18 :PRINT "                                 "

3090   SOUND 65,10

3130   LOCATE 51,0 :PRINT "POINT =";PNT;

3160   LOCATE 14,0 :COLOR C5 :PRINT "Time's up!"
3170   FOR I=1 TO 30 :SOUND 82,3 :SOUND 0,2 :NEXT I

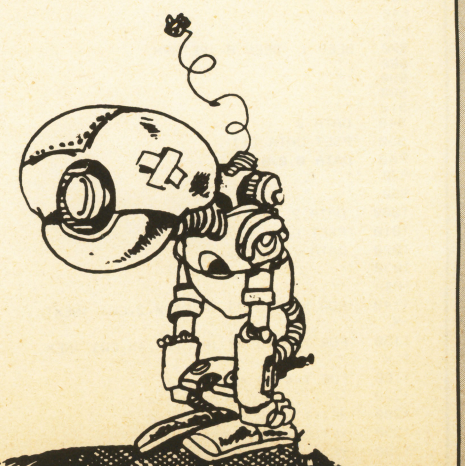
3220   SOUND 40,2

3240   LOCATE 58,0 :PRINT PNT;

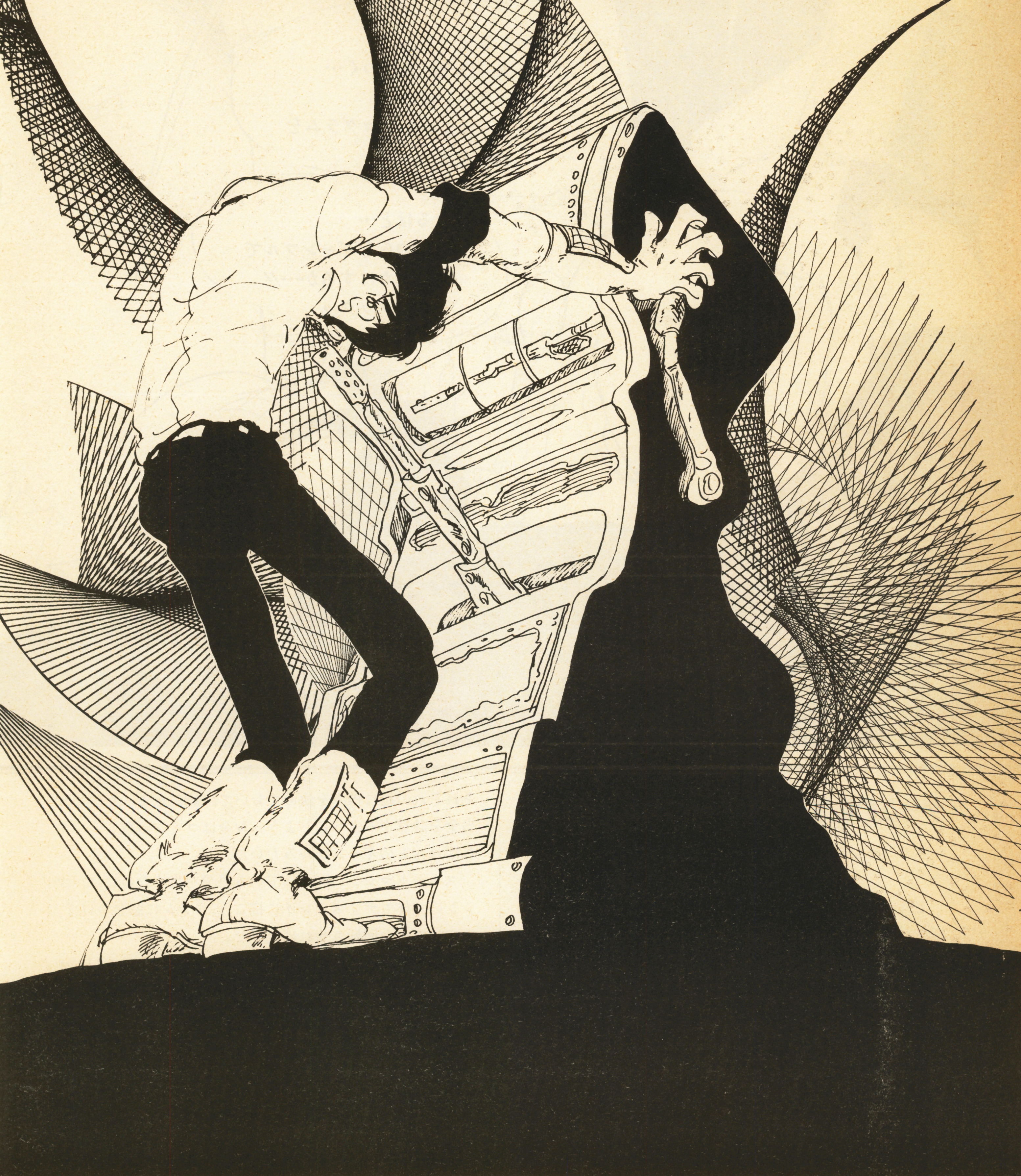
3300   LOCATE 20,0 :PRINT RT;

3320    IF RT=0 THEN ET=4:GOTO 2430:'Ending

3330    IF RT<=9 THEN SOUND 75,5
```



第3章 RUNだ!! 乱 スロットボンバー





ゲームの構成 フローチャートを作る

スロットマシンで乱たちに挑戦とは意外な展開になってきた。

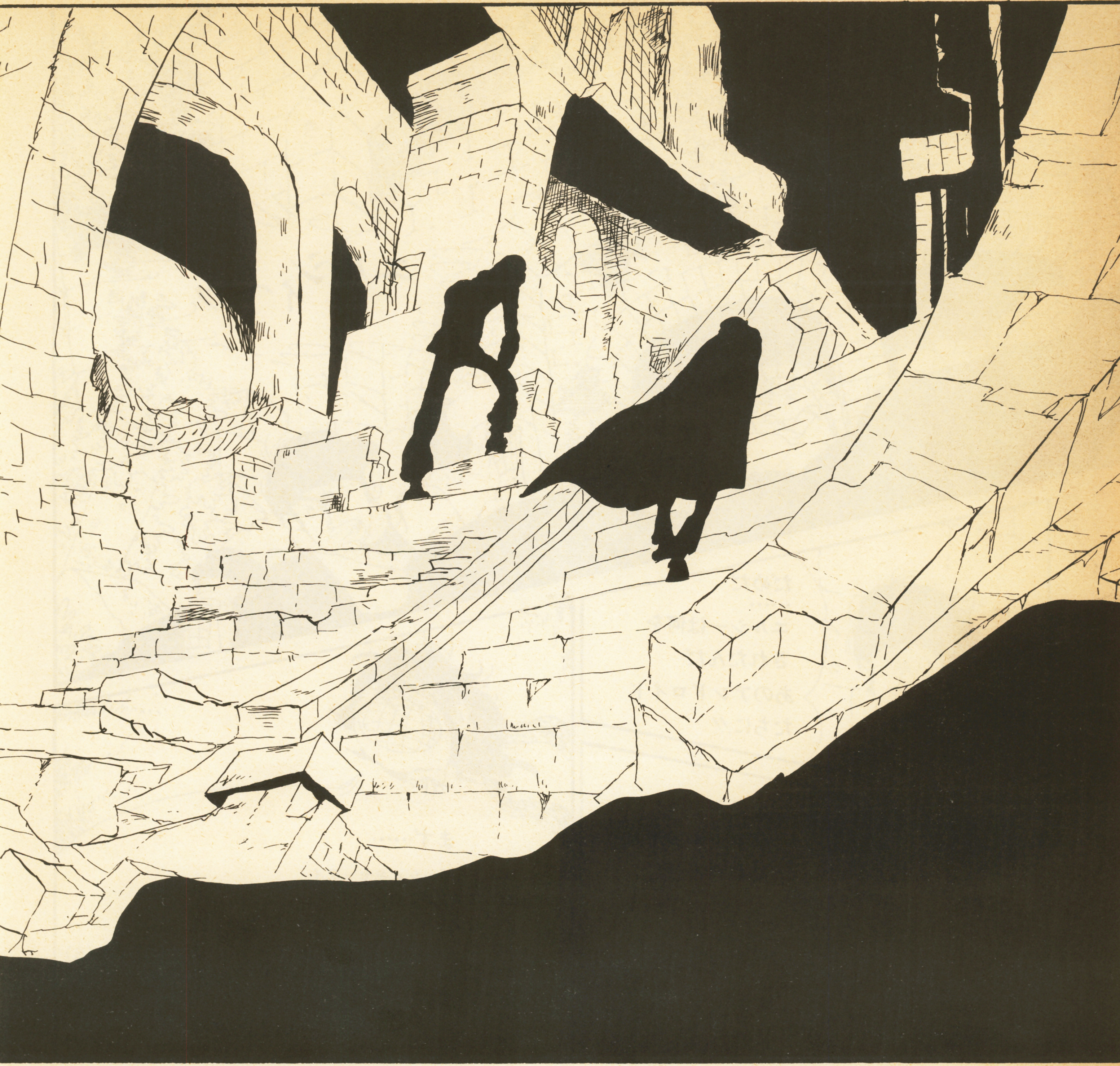
ところで，スロットマシンといえば，絵がくるくるまわってレバーを押せばピタッと止まるというのが普通だ。この絵のまわり方の"見せ方"がどうやらこのゲームのポイントとなりそうだね。ここがアイディアの見せどころ。難しいテクニックを使わなくても，何とかなる方法はないものだろうか？　ちょっと首をひねってみよう。プログラムはアイディアが勝負だ。

それからスロットマシンというからには配当金も欲しい。それぞれの絵に倍率を決めておいて，絵が止まった時に配当金を計算するようにしよう。お金はコインの数で表示すればほんとのギャンブルみたいだ。

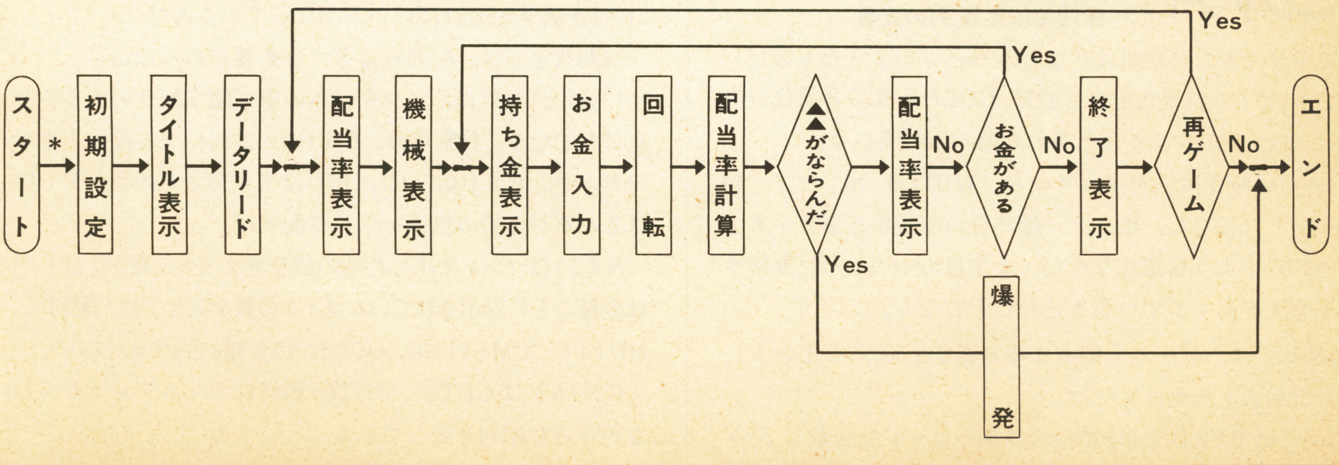
スロットマシンに向かっている気分を出すためには，マシンの本体もディスプレイに描こう。マシンに窓をつけて，その中で絵を回転させれば完璧だ。

資金があるうちはいくらでもゲームを続けられるが，オケラになったら終わりだ。ここでゲームオーバーということにしよう。スリルを加えるために，ある絵で止まったらマシンが大爆発を起こすようにしたらおもしろそうだ。

ゲームの流れは右のフローチャートのように組み立ててプログラムを組んでいこう。



スロットボンバーのフローチャート（プログラムのおおまかな構成）



＊ フローチャートの流れを表す矢印を「流れ線」という。



イニシャライズ PART 1 画面構成と配列の宣言

アルゴの考えた初期設定は右のページのようなものだ。

110行では，背景色を青に設定。COLOR 文の背景色の指定は，次のCLS 文が実行された時に実行される。

120行は画面構成を決めている。WIDTH 40，25で，40字モードに設定だ。横40字，縦25行の原稿用紙をディスプレイに描くような意味なんだ。ます目の1つ1つに文字やキャラクタを1つずつ書き込むことになる。

150行では，ゲームを始める時の資金を500にしている。持ち金を表す変数はMだね。

180行の READ 文は210行のカラーコードを変数に読み込んでいる。プログラムの中の色指定を変更したい時には，210行を書きかえればいいようにしているんだ。

初期設定では，配列宣言をする必要があったね。

130行と140行は，プログラムの中で使う配列の大きさを宣言している。H$ (10)，CH (10) は回転する絵の上半分の形と色，L$ (10)，CL (10) は下半分とその色，P (10) はそれぞれの絵の倍率を入れる配列だ。

N (2) はマシンを止めた時の絵を覚えておく配列，M (4) は金種ごとに表示されているコインの数を覚えておく配列だ。MD (4) と CM (4) には，そのコインの金額と色をしまっている。

G%(5) は GET@，PUT@ の時にディスプレイから絵を読み込む配列を表している。




```
100 REM ---------- ｼｮｷｾｯﾃｲ ----------
110  COLOR ,0 :CLS
120  WIDTH 40,25 :CONSOLE 0,25,0,0
130  DIM H$(10),L$(10),CH(10),CL(10),G%(5)
140  DIM P(10),N(2),M(4),MD(4),CM(4)
150  M=500
160 '
170 REM ---------- ｶﾗｰ ﾖﾐｺﾐ ----------
180  READ C1,C2,C3,C4,C5,C6,C7
190 '
200 REM ---------- ｶﾗｰ ﾃﾞｰﾀ ----------
210  DATA 1,2,3,4,5,6,7
220 '

420 REM ---------- ｵｶﾈ ｶﾗｰ & ｷﾝｶﾞｸ ﾘｰﾄﾞ ----------
430  FOR I=0 TO 3
440   READ MD(I),CM(I)
450  NEXT I
460 '
470 REM ---------- ﾃﾞｰﾀ ﾘｰﾄﾞ ----------
480  N=0
490  READ CH(N)
500 '
510  READ H$(N),CL(N),L$(N),P(N),CH(N+1)
520  N=N+1
530  IF CH(N)<>0 THEN 510
540 '
550  LOCATE 20,23 :PRINT "ﾅﾆｶ ｷｰｦ ｵｼﾃ ｸﾀﾞｻｲ";
560  IF INKEY$="" THEN 560
570 '

2930 REM ---------- ｷﾝｶﾞｸ & ｲﾛ ﾃﾞｰﾀ ----------
2940  DATA 1000,1, 100,2, 10,3, 1,4
2950 '
2960 REM ---------- ﾃﾞｰﾀ ----------
2970  DATA 4,╭╭
2980  DATA 2,●●,1.5
2990  DATA 4,▲
3000  DATA 5,└┘, .1
3010  DATA 2,♥┐
3020  DATA 1,└♠, .5
3030  DATA 4,╭╭
3040  DATA 2,♥♥, .2
3050  DATA 6,*┐
3060  DATA 6,└*, .4
3070  DATA 5,●●
3080  DATA 3,╰╰, .2
3090  DATA 2,♥◥
3100  DATA 2,◣♥, .3
3110  DATA 5,▲
3120  DATA 5,▲,  0
3130  DATA 0
```



スロットマシンを描く PRINT文とキャラクタを使う

アルゴとキルが変身したスロットマシンが現れた。なんでもプログラムを使って行動する二人のことだ。スロットマシンの形だってプログラムで作ったに違いない。そのプログラムをのぞいてみよう。右がリストだ。PRINT 文とキャラクタを使って描いていることがわかるだろう。

780行で，CLS を実行して画面をおそうじ。それからスロットマシンが現れるようにしている。

790行は，ディスプレー上で，上から2行分だけあけて描くようにするためだ。

色をつけたらもっと迫力が出そうだ。

```
770 REM-------- ｷｶｲ ﾋｮｳｼﾞ --------
780  CLS
790  PRINT :PRINT
800  PRINT"            |     |"
810  PRINT"  ┌─────────┴─────┴┐"
820  PRINT"  │                │"
830  PRINT"  ╞════════════════╡"
840  PRINT"  │                │  ███"
850  PRINT"  │┌──────────────┐│  ███"
860  PRINT"  ││ ──────────── ││  ███"
870  PRINT"  ││ │██││██││██│ ││  ███"
880  PRINT"  ││ ├██┤├██┤├██┤ ││  │ │"
890  PRINT"  ││ ├██┤├██┤├██┤ ││  │ │"
900  PRINT"  ││ │██││██││██│ ││  │ │"
910  PRINT"  ││ ──────────── ││  │ │"
920  PRINT"  │└──────────────┘│  │ │"
930  PRINT"  │                │  │ │"
940  PRINT"  ╞════════════════╡  │ │"
950  PRINT"  │                │  │ │"
960  PRINT"┌─┴────────────────┴─┐│ │"
970  PRINT"│                    ││ │"
980  PRINT"│                    ├┤ │"
990  PRINT"│                    ││ │"
1000  PRINT"│                    ││ │"
1010  PRINT"│                    ├┤ │"
1020  PRINT"└────────────────────┘└─┘";
```



こんな時にはPAINT文を使えたね。PAINT文は

PAINT：ペイント／色をぬるという意味

《役目》色をぬる

《使い方》PAINT（*x*，*y*），A，B*

座標（*x*，*y*）からBのカラーコードで描かれた境界までをAの色でぬりつぶす。

という命令だった。左のスロットマシンはC７という変数に読み込んだカラーコードで描いたから，BをC７にして，ぬりたい色をAで指定してやればいいね。

マシン全体は青（C１）に，細い横の線は赤（C２）にしてみよう。右のようなプログラムにすればいい。

ただ，この場合，PAINT文の使えないマイコンもある。どこまで色をぬるのかの境界が，ドットで描かれていないと，境界として扱わずに画面いっぱいにその色をぬってしまう機種もあるんだ。

左のプログラムではキャラクタでマシンを描いているから，レベル３やパソピアなどのPAINT文では色をつけられない。LINE文でマシンを描いておくか，色をつけるのをあきらめるかしよう。

もともとPAINT文のないPC 8001やMZ 700のようなマイコンもある。やっぱり色はつけられないね。

```
1040 REM ---------- ｷｶｲﾆ ｲﾛｦ ﾂｹﾙ ----------
1050  PAINT (50, 30),C1,C7
1060  PAINT (50, 43),C2,C7
1070  PAINT (50, 47),C1,C7
1080  PAINT (50,131),C2,C7
1090  PAINT (50,135),C1,C7
1100  PAINT (65,150),C1,C7
```

*　パソピアではPAINT文の色の境界色は１個だけ。



タイトルを描く　CIRCLEでサクランボを描く

キルやアルゴが乱たちの注意をひくためにしたように，ゲームを始める前に，華々しくタイトルを表示すれば，プレーヤーをゲームに引き込むことができる。マイコンゲームを作る時には，各人，凝ったタイトルの描き方を考えてみよう。

右のリストがタイトルを描くプログラムだ。

240行で背景色とタイトルの色を，250行ではタイトルを表示する位置を指定している。タイトルは，エレクトロマインと同じようにPRINT文とキャラクタで描こう。

310行のLINE文は，白いワクを描くためのものだ。直線でラインをひくためにはPSETが必要だったね*。

タイトルの下のサクランボは，340行から400行までで描いている。CIRCLE文でサクランボの輪郭を描き，PAINTで塗っているんだ。

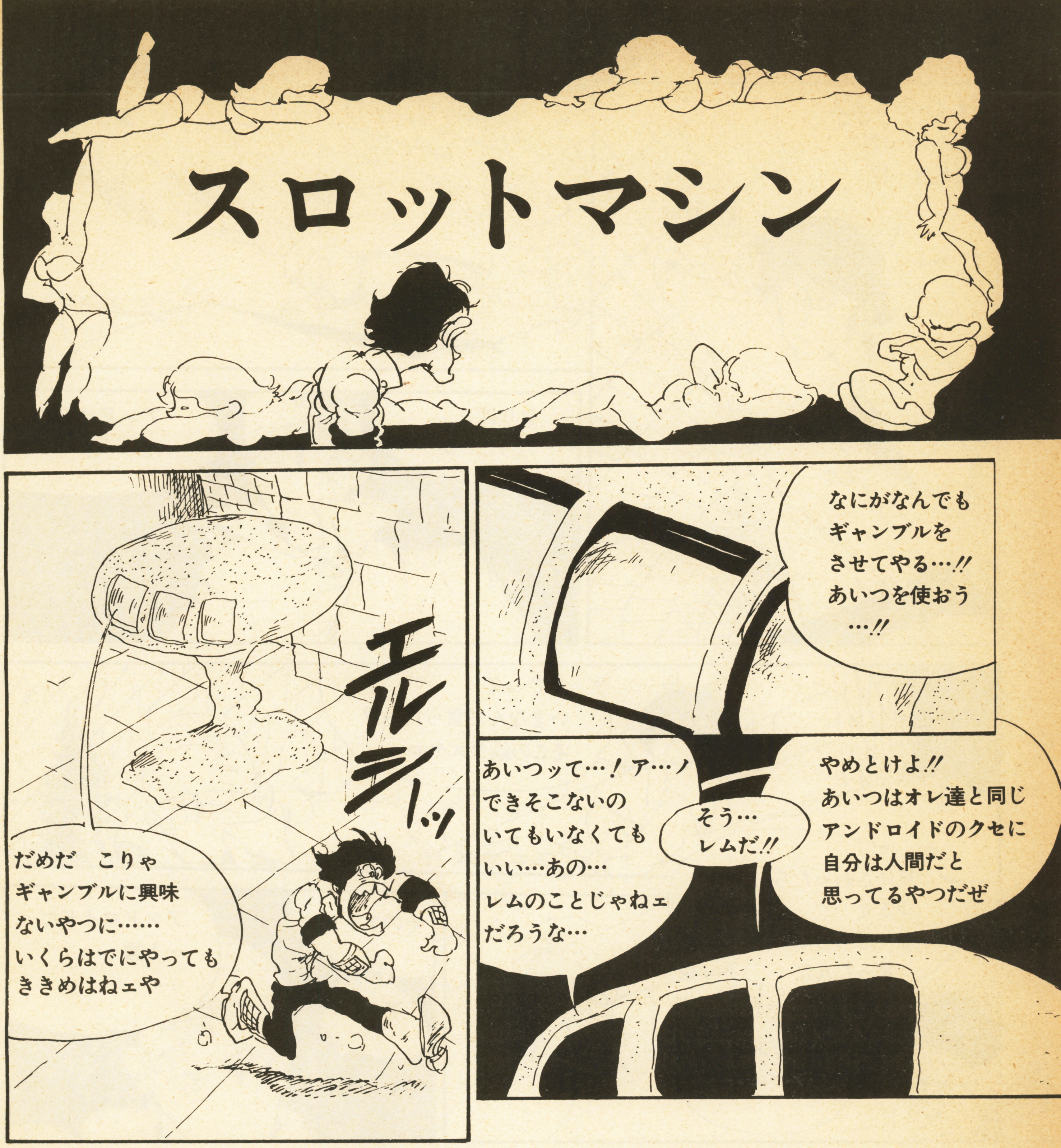
CIRCLE：サークル／円という意味

《使い方》CIRCLE（中心の座標），半径，色指定した座標を中心とする円を描く。

このCIRCLE文は64ページでも詳しく説明しているから，あわせて参照しよう。

CIRCLEやPAINTがないマイコンなら，タイトルと同じようにPRINT文とキャラクタで描いてみよう。高校生以上はsin（サイン）cos（コサイン）を使って円を描くテも

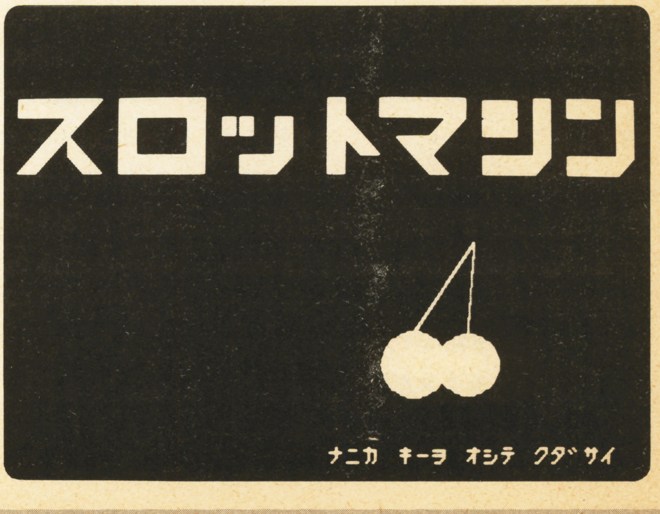
* MZ 700にはLINE文がない。FOR～NEXTとPSETで描こう。



考えてみよう。

マルを描いて赤くぬる。そして白い線を1本。これだけでも十分それらしく見えるね。ゲームにはちょっとしたデザイン感覚も必要というわけだ。

```
230 REM ---------- ﾀｲﾄﾙ ----------
240  COLOR C7,C1 :CLS
250  LOCATE 0,5
260  PRINT "
270  PRINT "
280  PRINT "
290  PRINT "
300 '
310  LINE (0,0)-(639,199),PSET,C7,B
320 '
330 REM ---------- ｻｸﾗﾝﾎﾞ ----------
340  CIRCLE(400,150),30,C2
350  PAINT (400,150),C2,C2
360  CIRCLE(450,150),30,C2
370  PAINT (450,150),C2,C2
380  CIRCLE(450,150),30,C1
390  LINE (455,100)-(400,137),PSET,C4
400  LINE (455,100)-(450,137),PSET,C4
410 '
```





始動準備OK BEEP 0 BEEP 1

キルとアルゴのスロットマシンはもうスタンバイOKだ。せっかくディスプレイにマシンを描いても，動かなくてはつまらない。チョッと工夫して，絵がまわり始める前にマシンがゴトゴト始動するようにしたらどうだろう。

マシンにお金を入れたら，スロットマシンの窓のふたがしまり，ウデが縮んで伸びるようにしてみよう。

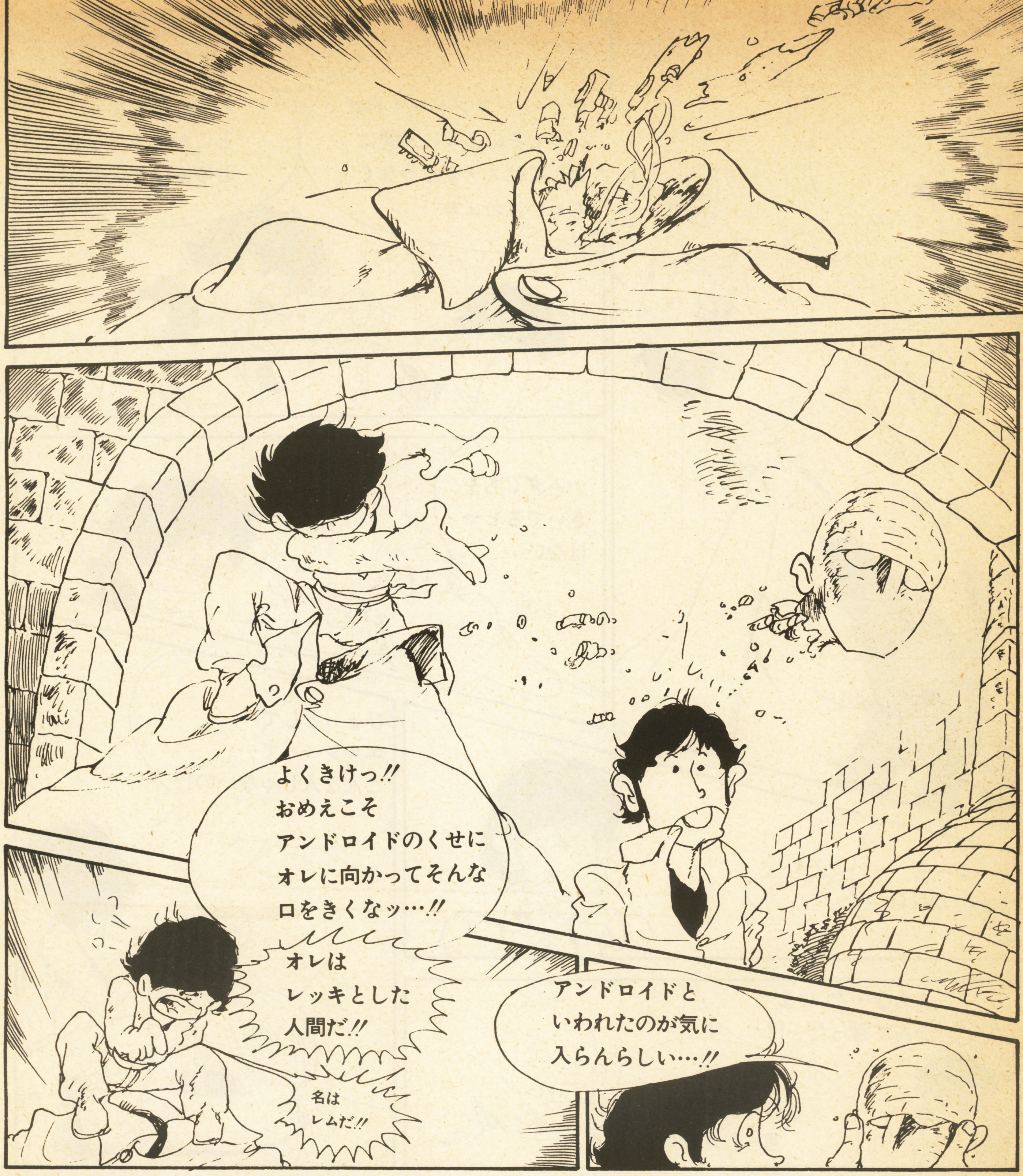
LOCATEを使って，マシンの窓の位置にふたのイメージを出すキャラクタを描けば，ふたをした感じになるね。ふたの色は白だ。

1290行から1340行のループは，1回まわるごとにLOCATEのx座標が変化し，水平方向に並んだ3つの窓がひとつずつしまるようにしている。

1370行からが，ウデを縮めて伸ばすプログラムだ。1370行から1400行のループは，1回まわるごとに，ウデの上をキャラクタひとつ消して（スペースを描いて），下にひとつキャラクタを描く。こうすれば縮む感じが出るね。

FOR～NEXTループの初期値が4になっているね。これは，変数YをループのΦで実行させるLOCATE文のy座標に使うためだ。ループを4から13までの10回まわすという他に，縮める場所を指定する意味があるんだ。

縮んだところで音を出そう！　音を出すにはBEEPという命令を使おう。



BEEP：ビープ／ビーという警笛を鳴らすという意味

《使い方》BEEP　1　または　0*

BEEP 1で音を出し，BEEP 0で止める。

1430行のループは，音の鳴っている時間を決めている。1420行から1440行のループの中で音は10回鳴るワケだ。

1460行から1490行のループは，1370行からのループと逆の方法でウデを伸ばしている。

ウデの伸び縮みには，窓をしめたのと同じLOCATE文とFOR～NEXT文を使っている。比べてみよう。

水平方向に仕事をさせたい時は，LOCATEの座標のxを，FOR～NEXTの変数でかえるように，垂直方向に仕事をさせたい時はyをかえるようにする。FOR～NEXTループの利用法として，いろいろに使えるテクニックだ。覚えておくと便利だよ。

```
1270 REM ----------- ﾏﾄﾞﾆ ﾌﾀｦ ｽﾙ -----------
1280  COLOR C7
1290  FOR I=0 TO 2
1300   LOCATE 6+I*4, 9 :PRINT "■■"
1310   LOCATE 6+I*4,10 :PRINT "■■"
1320   LOCATE 6+I*4,11 :PRINT "■■"
1330   LOCATE 6+I*4,12 :PRINT "■■"
1340  NEXT I
1350 '
1360 REM ----------- ｳﾃﾞﾉﾊﾞｼ -----------
1370  FOR Y=4 TO 13
1380   LOCATE 22,Y   :PRINT "   "
1390   LOCATE 22,Y+4 :PRINT "■■■"
1400  NEXT Y
1410 '
1420  FOR I=1 TO 10
1430   BEEP1 :FOR J=1 TO 5 :NEXT J :BEEP0
1440  NEXT I
1450 '
1460  FOR Y=13 TO 4 STEP -1
1470   LOCATE 22,Y+4 :PRINT "| |"
1480   LOCATE 22,Y   :PRINT "■■■"
1490  NEXT Y
```

*　一定時間だけ音を出すBEEPという命令もある。



イニシャライズ PART II READ～DATA

初期設定のリストで，まだ説明していないところがあったね。もう一度，その部分のリストを見てみよう。

```
420 REM ---------- ｵｶﾈ ｶﾗｰ & ｷﾝｶﾞｸ ﾘｰﾄﾞ ----------
430  FOR I=0 TO 3
440   READ MD(I),CM(I)
450  NEXT I
460 '
470 REM ---------- ﾃﾞｰﾀ ﾘｰﾄﾞ ----------
480  N=0
490  READ CH(N)
500 '
510  READ H$(N),CL(N),L$(N),P(N),CH(N+1)
520  N=N+1
530  IF CH(N)<>0 THEN 510
540 '
550  LOCATE 20,23 :PRINT "ﾅﾆｶ ｷｰｦ ｵｼﾃ ｸﾀﾞｻｲ";
560  IF INKEY$="" THEN 560
570 '
```

左の READ 文で下のデータを読み込んでいるんだ。よく見ると，読み込み方が少し違うね。

```
2930 REM ---------- ｷﾝｶﾞｸ & ｲﾛ ﾃﾞｰﾀ ----------
2940  DATA 1000,1, 100,2, 10,3, 1,4
2950 '
2960 REM ---------- ﾃﾞｰﾀ ----------
2970  DATA 4,,,
2980  DATA 2,●●,1.5
2990  DATA 4,▲
3000  DATA 5,凵, .1
3010  DATA 2,♥┐
3020  DATA 1,└♠, .5
3030  DATA 4,,,
3040  DATA 2,♥♥, .2
3050  DATA 6,*┐
3060  DATA 6,└*, .4
3070  DATA 5,●●
3080  DATA 3,ヽヽ, .2
3090  DATA 2,♥◥
3100  DATA 2,◣♥, .3
3110  DATA 5,▲
3120  DATA 5,▲,  0
3130  DATA 0
```



コインの色と金額を読み込んでいるREAD文を見てみよう。配列の中のIは，変数をFOR～NEXTで設定している。同じ金種のコインの金額と色は，同じ添字の配列にしまっておいたほうが，プログラムの中でコインを扱う時扱いやすい。それで，1回ループをまわすあいだに金額と色をワンセットで読んでしまおうというワケだ。

回転する絵に関するデータは，FOR～NEXTではなく，IF～THENで作ったループで読んでいる。これは，キミたちが絵の種類をもっと増やしたいと思った時，3130行より前にいくつでもDATA文を加えられるようにしたんだ。IF～THENのループは，3130行を読むことで抜けられるからね。この判断は530行でしている。FOR～NEXTを使っていると変数もかえなくてはならない。

もうひとつ，490行で2970行の“4”だけを読んで，その次から，1回のループで次の絵のDATA文の頭までを読むループに入っているのはなぜだろう。ループを抜ける判断材料にするデータを1つだけにするためなんだ。

自分で絵のデータを付け加えたい時には，グラフィックなキャラクタを使おう。GRAPHキーと一緒にキミの好きなキャラクタの入っているキーを押せばいい。デザイン感覚を生かそう。

キャラクタのキーボード上の配列は，マイコンの種類によって違う。キミの使っているマイコンのマニュアルを参照しよう。キャラクタ表は必ずのっているはずだ。



お金の入力 PRINT USING ビット

とうとう乱もスロットマシンに興味を示した。アルゴたちがさっき渡したコインを元手に,マシンへ向かっていく。コインをマシンに入れてゲーム開始だ。

```
1120 REM ------- ｵｶﾈ ﾆｭｳﾘｮｸ -------
1130  GOSUB 2320
1140 '
1150  COLOR C7
1160  LOCATE 26,1 :PRINT USING "ﾓﾁｷﾝ ######";M
1170  LOCATE 26,5 :INPUT "ｶｹｷﾝ "; CIN$
1180  CIN=VAL(CIN$)
1190  IF CIN>M OR CIN<=0 THEN BEEP :GOTO 1170
1200  M=M-CIN
1210 '
1220  LOCATE 1,20 :PRINT "                    "
1230  LOCATE 26,1 :PRINT USING "ﾓﾁｷﾝ ######";M
1240  LOCATE 12,2 :PRINT USING "|#####|";CIN
1250  GOSUB 2320
1260 '
```

まず2320行からのサブルーチンを呼んでいる。このサブルーチンは,お金をコインの形で表示するものだが,これについては後で説明しよう。

1160行で持ち金を描く。はじめの持ち金は500だね。87ページで150行のリストを見よう。

見慣れないPRINT文があるね。

PRINT USING*:プリント ユージング/USING は use(使う)のing形

《使い方》PRINT USING "##.##";数値または文字(文字の場合は#でなく&を使う)

指定した表示のパターンで書き出す。

1160行のパターンだと,Mが小数の時,小数点以下を四

＊ PRINT USINGは一覧表の作成なんかに便利。



捨五入して表示する。

いくらお金をかけるかは，1170行の実行で要求される。持ち金より多い額をかけようとすると，音が鳴って，もう一度かけ金の入力を要求してくる。1190行だ。

1200行では，持ち金からかけ金をひいて，かけ金を入力した時点の持ち金を計算している。この時のお金の状態を表示し直すのが1230行から1240行だ。

1230行と1240行で，さっき書いた持ち金とかけ金をそれぞれ書きかえているんだ。

ここでまた，コインを表示する2320行からのサブルーチンを呼んで，コイン表示も書きかえている。

1220行は，マシンの一部を消しているんだけど，なぜそうするのかは後でまた説明しよう。

ところで，ビット*という言葉が出てきた。上では円やドルのようなお金の単位に使っているようだけど，もともとはコンピュータに関係した言葉なんだ。コンピュータも電気で動いているので，スイッチが入っているかいないか，電圧が高いか低いかという区別はできるわけだ。これをたくみに0と1の2通りの状態になぞらえて計算などをやろうとしているのが今のコンピュータ。そして，ある状態が0か1かを示す単位がビットというわけだ。

乱たちの時代にはコンピュータの重要度はますます高くなるから，ビットというのがお金の単位にまでなったのだろうね。

* ビット＝bitと書いて，これは2進数という英語 binary digit の省略形だ。



配当率の表示 TAB PRINT USING

なんと! 乱がチャレンジする気になったのは，スロットマシンではなくてポーカーゲーム。アルゴたちも負けてはいない。配当率を表示して興味をひいてきた。

470行から530行のREAD文で読み込んだ変数を使って表示させている。

データを配列に読み込む時，ひとつの絵に関するデータは，同じ添字で読むようにしたね。だからここでも，650行から720行の1つのFOR～NEXTループで，絵と倍率の両方を表示できるんだ。

Nには，絵の数が入っている。(94ページ480行を参照)

650行から720行のループは，そのNを使って0回からN－1回までのN回まわすようにしているんだ。

660行と670行にTABという命令がでてくるね。

TAB：タブ／tabulator（表作成装置）の略

《使い方》PRINT TAB（ケタ数）；"文字"

指定したケタ数だけ行頭からあけて文字を表示する*。

670行では，H$(I) にしまわれている絵を，左から5文字あけた場所に描くように命令しているワケだ。

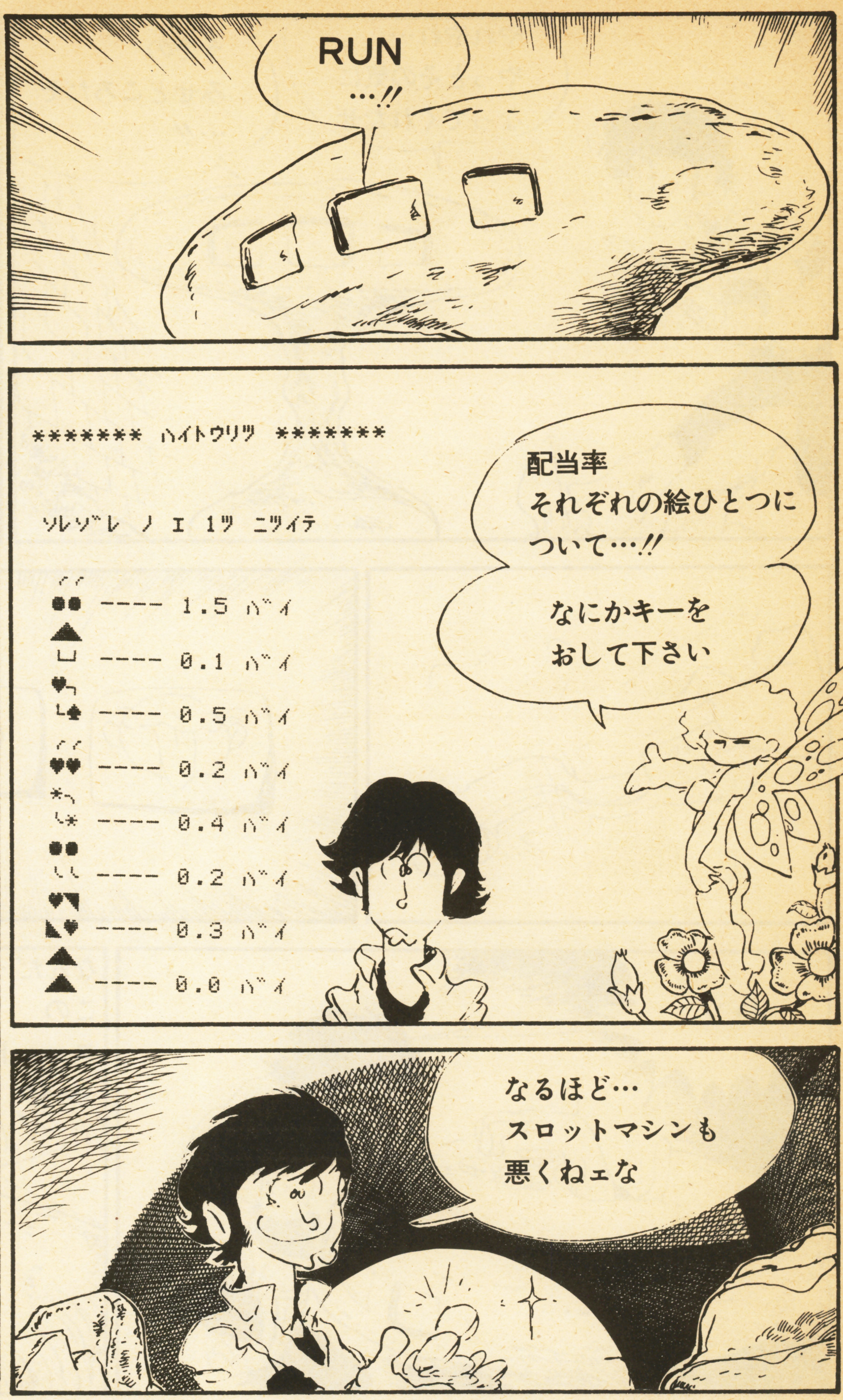
600行のように，スペースを文字列として入れても同じことができるけど，たくさんあけたい時なんかはTABのほうが便利だ。

690行を見よう。PRINT USINGでは，表示のパターン

＊ ケタ数や表示の位置などの指定は，そのマイコンが何文字モードかによって調節する必要がある。

```
580 REM ----------
       ﾊｲﾄｳﾘﾂ ﾋｮｳｼﾞ
       -------------
590  COLOR C7,0 :CLS
600  PRINT
      "        *******
       ﾊｲﾄｳﾘﾂ *******"
610  PRINT
620  PRINT
630  PRINT
     " ｿﾚｿﾞﾚ ﾉ ｴ 1ﾂ ﾆﾂｲﾃ"
640  PRINT
650  FOR I=0 TO N-1
660   COLOR CH(I)
     :PRINT TAB(5);H$(I)
670   COLOR CL(I)
     :PRINT TAB(5);L$(I);
680   COLOR C7
690   PRINT USING
      " ---- #.# ﾊﾞｲ";P(I)
700   BEEP
710   FOR J=1 TO 200
     :NEXT J
720   NEXT I
730 '


2960 REM ---- ﾃﾞｰﾀ ----
2970  DATA 4,╭╭
2980  DATA 2,●●,1.5
2990  DATA 4,▲
3000  DATA 5,└┘, .1
3010  DATA 2,♥╮
3020  DATA 1,└♣, .5
3030  DATA 4,╭╭
3040  DATA 2,♥♥, .2
3050  DATA 6,*╮
3060  DATA 6,╰*, .4
3070  DATA 5,●●
3080  DATA 3,╰╰, .2
3090  DATA 2,♥◥
3100  DATA 2,◣♥, .3
3110  DATA 5,▲
3120  DATA 5,▲,  0
3130  DATA 0
```



として，文字も一緒に書けるんだ。数に単位をつけて表示したい場合に使えるね。

一度に全部の絵について表示すると，めんどうくさくてよく読まなかったりする。1行表示するごとに，チョッと時間をおくようにしよう。710行のようなカラまわりの時間をつぶすFOR～NEXTループを入れてやればいい。

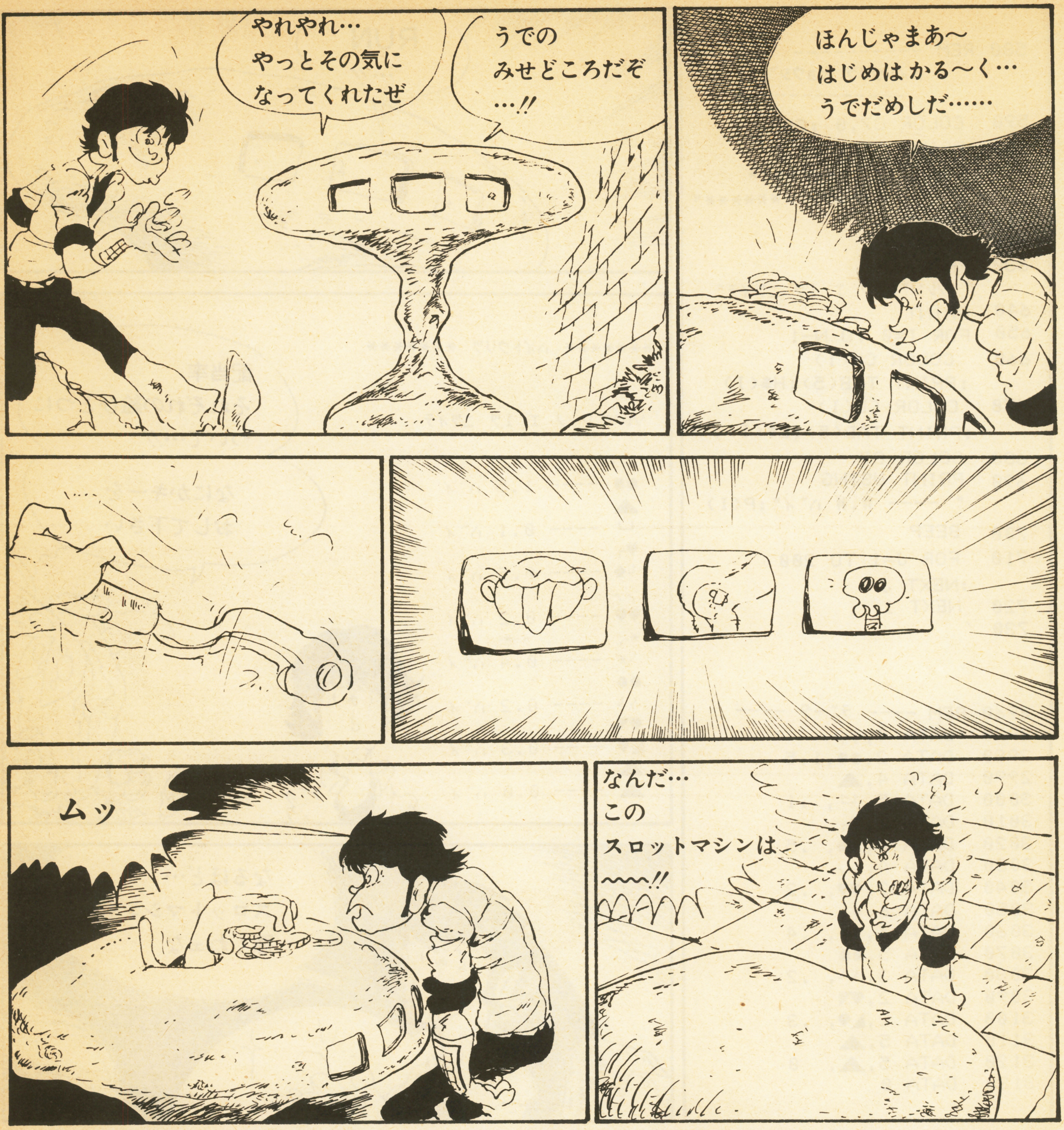
700行のBEEPは，BEEP 1，0とは違って，一定の時間だけ音を鳴らす命令だ。

FOR～NEXTのループを抜け出したら，740行で，次の画面に移るための合図を要求される。750行では，何か入力されるまで，待ちの状態にしておくために，1行だけのループを作っている。

どんな場合にどれくらいもうかるかを具体的に見せられて，乱もやっとやる気になってきたね。

アルゴたちの思惑にひっかからなければいいが……。

```
580 REM ---------- ﾊｲﾄｳﾘﾂ ﾋｮｳｼﾞ -------------
590  COLOR C7,0 :CLS
600  PRINT "        ******* ﾊｲﾄｳﾘﾂ *******"
610  PRINT
620  PRINT
630  PRINT " ｿﾚｿﾞﾚ ﾉ ｴ 1ﾂ ﾆﾂｲﾃ"
640  PRINT
650  FOR I=0 TO N-1
660   COLOR CH(I) :PRINT TAB(5);H$(I)
670   COLOR CL(I) :PRINT TAB(5);L$(I);
680   COLOR C7
690   PRINT USING" ---- #.# ﾊﾞｲ";P(I)
700   BEEP
710   FOR J=1 TO 200:NEXT J
720  NEXT I
730 '
740  LOCATE 20,24 :PRINT "ﾅﾆｶ ｷｰｦ ｵｼﾃ ｸﾀﾞｻｲ";
750  IF INKEY$="" THEN 750
760
```



絵の回転 PART Ⅰ

INT, RND, INKEY$

コインを入れて，いよいよ絵が回転しはじめるぞ。どんなプログラムを作ったら回転して見えるだろうか。

まず，2970行以降に用意された絵のうち，どれから回転を始めるか，決めなくてはならない。いつも同じ絵から始まったのでは，パターンを覚えればねらいやすくなってしまうからね。こんな時はRNDを使おう。

RND：ランダム／不規則という意味のrandomの略
《使い方》RND（式）
0から1までのあいだの数を不規則に発生する。式が正の値なら同じ乱数の系列の正の乱数を与え，0なら前に発生した乱数を，負なら新しい乱数の系列を作る*。

1530行では，RND（1）で0～1の数を発生させ，Nつまり絵の数をかけて整数化することで，絵をしまってある配列の添字を指定しているんだ。整数化？　INTだ。

INT：インテジャー／整数という意味のintegerの略
《使い方》INT（式）
式で表す数値を，その数を越えない最大の整数に直す。

乱数は0～1未満の小数で発生するから，添字として使うには整数化する必要があるんだ。

ここまでで求めた添字は変数Jに読み込んでおこう。

実際に絵を回転させる命令はサブルーチンにしよう。このサブルーチンについては，また後で説明しよう。このサ

* 前にRANDOMIZE命令をすることで，実行させるたびに違った系列の乱数を得ることができる。



ブルーチンの中で，Ｊはだんだん増えていく。

スロットマシンの絵の回転は，止める合図をするまで続かなくてはならない。何か入力することで抜け出られるようなループを作って，その中で回転のサブルーチンを呼ぶようにしよう。1540行がそれだ。

合図してピタッと止まるのでは，ねらいやすくてゲームがつまらない。合図してから，回転がだんだんゆっくりになっていって止まるようにしよう。1550行から1580行までのループがそれだ。

1540行を抜けた後，1550行で，合図の後，何回絵をまわすかを，RNDで求めている。１回ループをまわすごとに1570行のＫの値が増えて，カラまわりの回数が増え，回転速度がゆっくりになっていくわけだ。

絵が止まったら，その絵をＮ(I)に読んでおこう。倍率の計算に必要だ。

さて，ここまでの流れで，1520行から1600行のFOR～NEXTループを１回だけまわったことになる。３回まわって３つの窓だ。

```
1510 REM ------- ウコ゛カス ! -------
1520  FOR I=0 TO 2
1530   J=INT(RND(1)*N)
1540   IF INKEY$="" THEN GOSUB 2160 :GOTO 1540
1550   FOR K=1 TO INT(RND(1)*N+3)
1560    GOSUB 2160
1570    FOR L=1 TO K*15 :NEXT L
1580   NEXT K
1590   N(I)=J
1600  NEXT I
1610 '
```



持ち金がなくなると終わり END処理

ギャンブルの常で，乱はオケラになってしまった。こんな時，カジノではディーラーから元手を借り入れるね。ゲームでもそうしよう。もちろん，持ち金がなくなったらギブアップもできるようにしたい。

持ち金が0になったら，この回のトライアルは終わりという意味で，マシンの窓に赤い“×”を描こう。1つの窓につけるたびに音を鳴らすとダメ押しの感じだね。1910行から2000行のループを1回まわると，“×”が1つだ。

3つの窓に“×”をつけ終わったらマシンを消そう。背景が黒だから，マシンを黒く塗ればいい。PAINT命令は，指定した色自身も境界とみなすから，黒い背景に囲まれたマシンを黒く塗るのに境界色の指定はいらないはずだ。でも，持ち金が0になる時の画面には黄色の文字が描かれているので（後で説明する），その文字を消さないために黄色を背景色に指定しているんだ。

マシンを消しながらまた音を鳴らそう*。ゲームの節目節目に音を鳴らしてやるのも，ゲームにめりはりをつけるテクニックの1つだ。

ここで，「モウイチド　ヤリマスカ？」と尋ねさせるようにしよう。2080行だ。

何もあせって決めることはない。キミがもう一度，トライすべきかどうかまよっているあいだ，2090行のループで

＊ パソピアでは，BEEPではなく，SOUNDという命令を使う。



マイコンはずっと待っていてくれる。

「今度こそ，もうけてみせるぞ！」と思うなら，Yかyを入れれば，また元手が500もらえるようにしよう。そして580行にもどって，かけ金の入力から始めるようにすればいい。これが2100行だ。

Nかnが入力されると，2110行を素通りしてゲーム終了だ。2110行は何だろう？　もう一度トライしたくて，Yを押したつもりで間違ったキーを押してしまった時に，終了するのではなく，入力待ちの状態にもどるようにしているんだ。間違いなんだから警告音を鳴らしてやろう。

このようなEND処理はプログラムの中で大変重要な部分だ。ヨーク理解しておこう。

```
1890 REM ------- オカネガ゛ナクテ オワリ -------
1900  COLOR C2
1910  FOR I=0 TO 2
1920   LOCATE 6+I*4, 9 :PRINT "  "
1930   LOCATE 6+I*4,10 :PRINT "╲╱"
1940   LOCATE 6+I*4,11 :PRINT "╱╲"
1950   LOCATE 6+I*4,12 :PRINT "  "
1960 '
1970   FOR J=1 TO 10
1980    BEEP1 :FOR K=1 TO 10 :NEXT K :BEEP0
1990   NEXT J
2000  NEXT I
2010 '
2020   PAINT (45,30),0,C6
2030   FOR J=1 TO 150
2040    BEEP1 :FOR K=1 TO 10 :NEXT K :BEEP0
2050   NEXT J
2060 '
2070  COLOR C7
2080  LOCATE 1,20 :PRINT "モウイチド゛ ヤリマスカ ? (Y/N)";
2090 A$=INKEY$ :IF A$="" THEN 2090
2100 IF A$="Y" OR A$="y" THEN M=500 :GOTO 580
2110 IF A$<>"N" AND A$<>"n" THEN BEEP :GOTO 2090
2120 '
2130 CLS
2140 END
```



スロットマシンの爆発

ここで，乱が組み込んだプログラムを解明してみよう。REM文をみると，どうやら爆発のサブルーチンらしい。

2660～2700行は，配当率の表示の欄に倍率を出す代わりに“……… !!”を表示する。そして，BEEPで警告音の発信だ。しばらく鳴らし続けるためにFOR～NEXTで時間つぶしをしているようだ。

そして，いよいよ爆発。爆発のシーンは四方八方ランダムに赤い放射線を描いてそれらしく見せている。こんな時はRND（ランダム）関数を使って，四方八方に飛び散る放射線の終点を決めよう。そして終点が決まったらスロットマシンの中心位置からＬＩＮＥ文で終点まで直線をひいてできあがり。ついでに，爆発音をBEEPで入れて仕上がりというわけだ。

ところでここのLINE文は，ふつうの線の代わりに■マークでひいているね。

ＬＩＮＥ $(x_1, y_1)-(x_2, y_2)$，PSET，色番号

ＬＩＮＥ $(x_1, y_1)-(x_2, y_2)$，キャラクタ，色番号

と２通りのLINE文があるわけだ。つまりPSETなら，

————————と線がひけるし，PSETの代わりに“A”とすれば，ＡＡＡ……ＡＡという文字が直線状に並んだ線がひけるということだ。もちろん斜めだってひける。あとは画面を消して，ハイおしまい。




```
2640 REM ---------- ﾊﾞｸﾊﾂ ----------
2650  COLOR C6
2660  FOR X=1 TO 16
2670   LOCATE X,20 :PRINT ".";
2680   BEEP1 :FOR J=1 TO 25 :NEXT J :BEEP0
2690  NEXT X
2700  PRINT " !!"
2710  BEEP
2720 '
2730  FOR I=1 TO 1000 :NEXT I
2740 '
2750  COLOR C2
2760  FOR I=0 TO 50
2770   X=INT(RND(1)*30) :Y=INT(RND(1)*25)
2780   LINE (10,10)-(X,Y),"■",C2
2790   BEEP
2830  NEXT I
2810 '
2820  FOR I=1 TO 50
2830   BEEP1 :FOR J=1 TO 10 :NEXT J :BEEP0
2840  NEXT I
2850 '
2860  LOCATE 0,24
2870  FOR I=0 TO 24 :PRINT :NEXT I
2880 '
2890  COLOR 7
2910  END
2920 '
2900  CLS
```

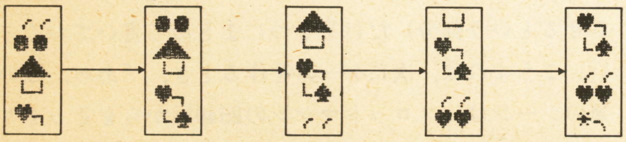
この爆発のシーンの作り方は，エレクトロマインの時と似ているね。このテクニックを１つ覚えれば，いろんなゲームで爆発のシーンを作ることができるわけだ。

こういうプログラムは，ふつうその機能（ここでは爆発させるという部分）だけを１つにまとめて書いておくといい。こうすれば，次にゲームを作るとき，「あっ，この爆発のシーンは，スロットマシンの2640行からをそっくり使えば簡単だ」なんてことになるわけだ。効率よくプログラミングできるだろう。そして，そのようなプログラムのブロックにはＲＥＭ文で注釈をつけるとさらにわかり易い。



絵の回転 PART II
アイディアの出しどころ

さて，いよいよ絵の回転を考えてみよう。アイディアの出しどころだ。絵を単に次々に表示するだけでは，何となく回転しているというイメージがわかない。そこで，こうしたらどうだろう。



上のように表示したい絵の前後に出てくる絵の下半分と上半分を一緒に表示してやるわけだ。こうすると見ての通り，連続的に変化して見えるだろう。さて，実際にこれをディスプレイに出すのに GET @, PUT @*を使ってみよう。

GET @，PUT @には4種類の形式がある。カラーコードを読むか読まないか，その中でも，キャラクタを読むかドットを読むか，の4種類だ。ここでは，キャラクタとカラーコードを読む形式を使う。

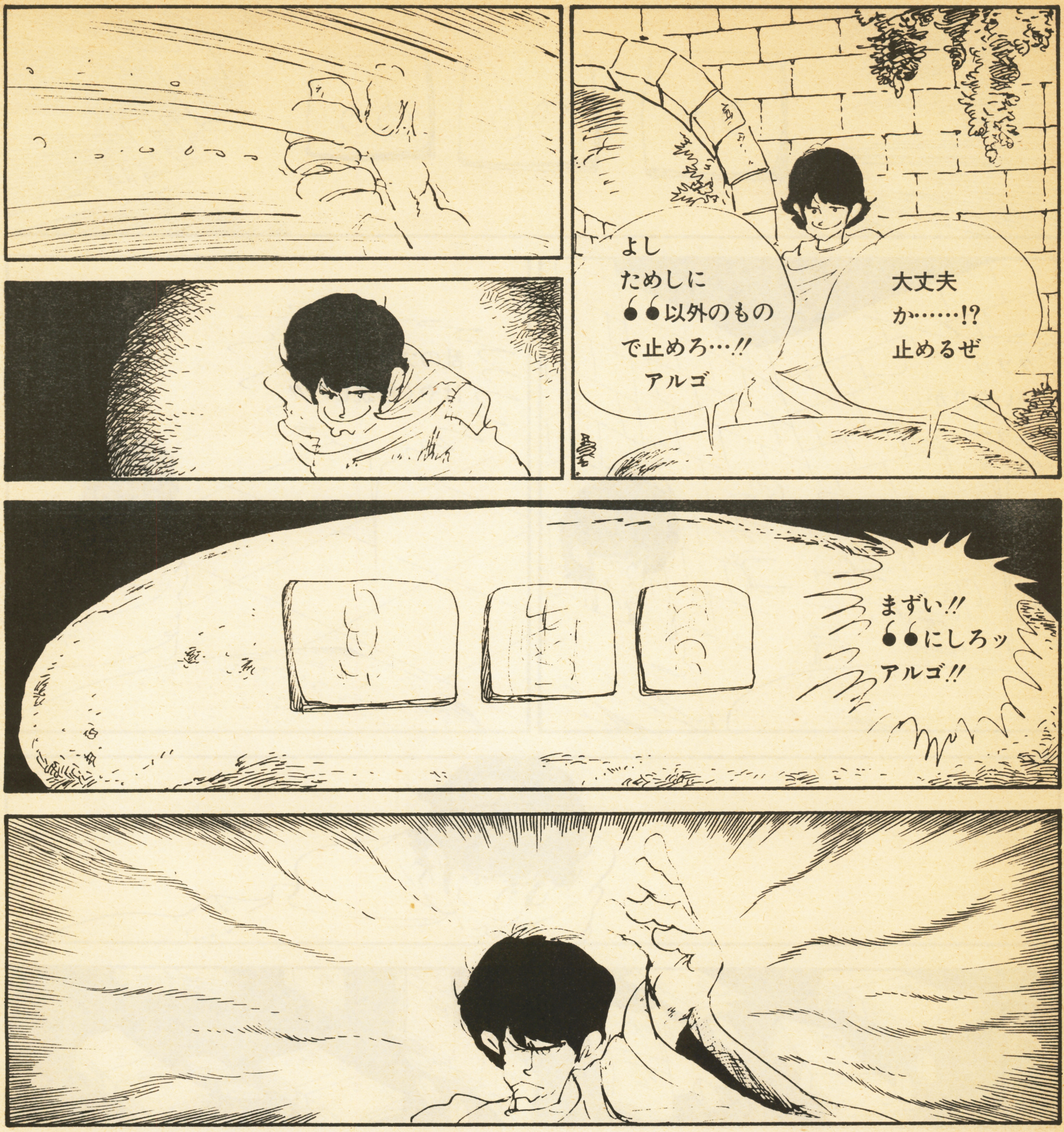
GET @，PUT @：ゲット・アット，プット・アット／get は得る，put は置く，という意味

《使い方》GET @A $(X_1, Y_1)-(X_2, Y_2)$，配列名

PUT @A $(X_3, Y_3)-(X_4, Y_4)$，配列名

GET @文で配列に読み込んだキャラクタとカラーコードを，PUT @文で指定する位置に表示する。位置の指定はキャラクタ単位の座標で表す。

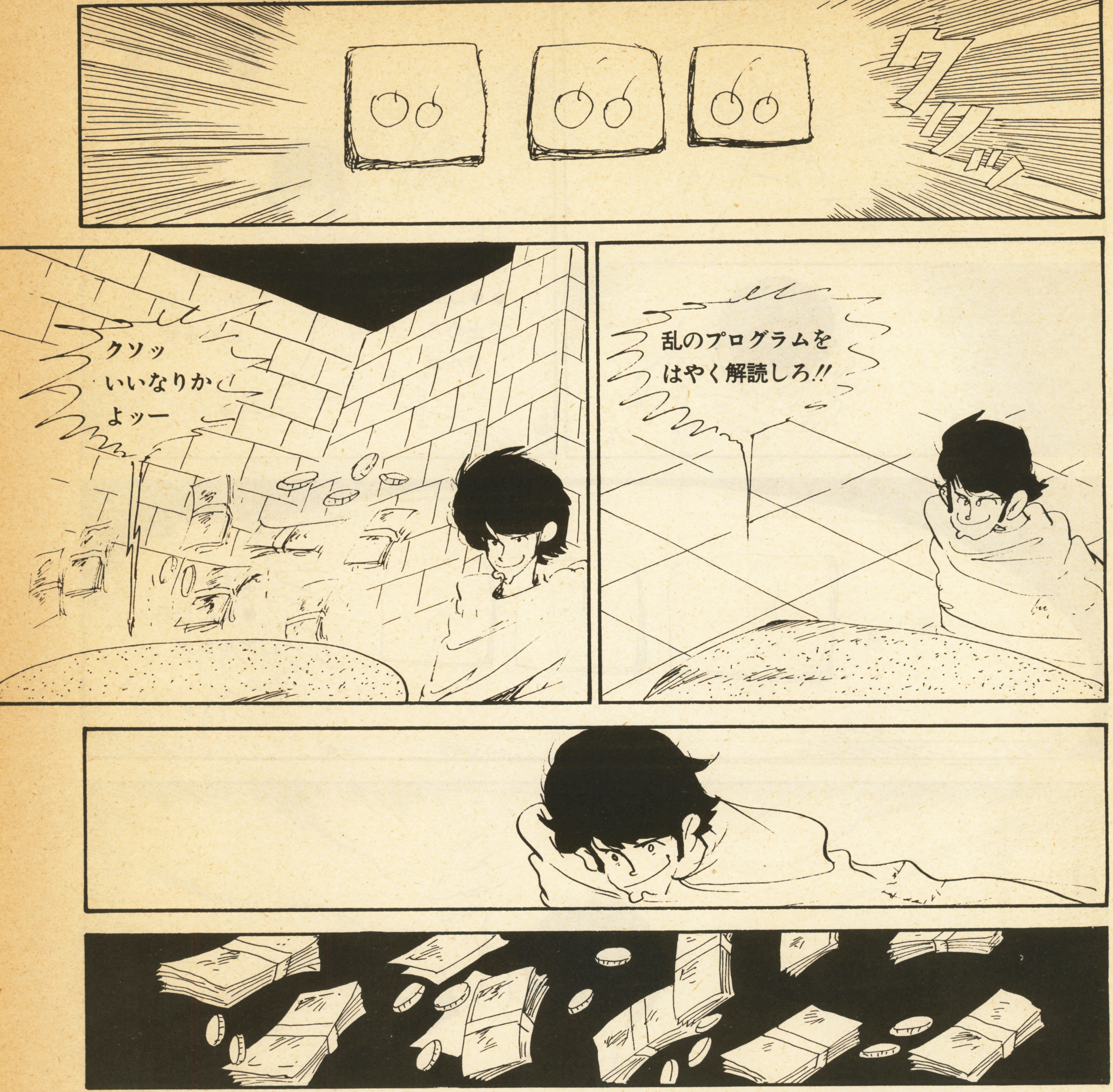
＊ GET @，PUT @のない機種は LOCATE と PRINT を使おう。



1530行（101ページ参照）でJの値を決め，1540行で右の回転サブルーチンを呼んでいる。1540行のループを1回まわるごとに2170行でJの値は増えていくわけだ。ところでこのJは，それぞれの絵の倍率が入っている配列の添字として使われる。だから，絵の数より大きくなってしまってはこまるんだ。そのために，MOD N（Nは絵の数だったね）によって0～N−1の範囲内を循環するようにしている。MODについては33ページを参照しよう。

1つの窓を横に4つに分け，下の3つにJを添字に持つ絵とJ+1の絵の上半分を表示する。それを配列G%に読み込んで（2190行），1つ上の位置に読み出す（2200行）。あいた部分に同じ絵の下半分を描く（2220行から2230行）。この時の下3つをまた配列G%に読み込む（2250行）。1つ上にずらして表示する（2260行）。あいた場所に次の絵の上半分を表示する（2280行から2290行）。メインルーチンへもどる（2300行）………というしくみだ。

```
2160 REM ------- カイテン -------
2170  J=(J+1) MOD N
2180  BEEP1 :BEEP0
2190  GET@A(6+I*4,10)-(7+I*4,12),G%
2200  PUT@A(6+I*4, 9)-(7+I*4,11),G%
2210 '
2220  LOCATE 6+I*4,12
2230  COLOR CL(J) :PRINT L$(J)
2240 '
2250  GET@A(6+I*4,10)-(7+I*4,12),G%
2260  PUT@A(6+I*4, 9)-(7+I*4,11),G%
2270 '
2280  COLOR CH( (J+1) MOD N)
2290  LOCATE 6+I*4,12 :PRINT H$((J+1) MOD N)
2300  RETURN
2310 '
```



お金の計算 同じ絵が出れば倍率が2倍に

トライアルが1回終わるごとに，配当率を計算し，手元にいくら持ち金があるかを計算するようにしよう。

配当率の変数はMULだ。各回の計算を始める時に，必ず0にもどしておく（1630行）。

1640行から1660行のループで，3つの窓で止まった絵の倍率の合計をだしている。これが配当率だ。N(I)は，止まった絵の番号を表している（1590行参照）。

♠が3つなら，爆発ルーチンへ飛ぼう（1680行）。

もし，同じ絵があったら，それぞれを2倍の倍率で計算することにしたらどうだろう。1700行から1720行でこれをやらせている。

同じ絵かどうかは，絵の番号が同じかどうかで判断させればいい。1つめの窓で止まった絵の添字はN(0)に，2つめはN(1)に，3つめはN(2)に入れてある（1520行から1600行を参照）。

同じ絵なら倍率も同じだから，一方の倍率を2倍したものを全体の倍率に加えれば，同じ絵の倍率を2倍として計算したのと同じことだ。1640行から1660行のFOR～NEXTループで計算した配当率MULに，2倍ぶんを加えてやればいい。これが1700行から1720行のIF～THEN文のTHEN以降に書かれた計算の意味だ。

3つとも同じ絵が出れば普通に計算した場合の3倍の配



当率だ。すごいもうけになるぞ！

配当率は整数に直しておこう。1730行だ。

1760行から1790行まででディスプレイに「…………」を描いて，配当率が0なら“スカ”，それ以外ならその数値を表示しよう（1800行）。

配当率が出たら，手持ちの金額がいくらになるか計算しよう。1820行だ。かけ金に配当率をかけたものを持ち金に加えて，新しい持ち金を出している。

ちょっと間をおいたら（1840行），かけ金の表示を消して（1860行），まだ持ち金が残っているなら，かけ金の入力要求にもどろう（1870行），再トライだ！

```
1620 REM -------- ｹｲｻﾝ --------
1630  MUL=0
1640  FOR I=0 TO 2
1650   MUL=MUL+P(N(I))
1660  NEXT I
1670 '
1680  IF MUL=0 THEN 2640
1690 '
1700  IF N(0)=N(1) THEN MUL=MUL+P(N(0))*2
1710  IF N(0)=N(2) THEN MUL=MUL+P(N(0))*2
1720  IF N(1)=N(2) THEN MUL=MUL+P(N(1))*2
1730  MUL=INT(MUL)
1740 '
1750  COLOR C6
1760  FOR X=1 TO 13
1770   LOCATE X,20 :PRINT ".";
1780   BEEP1 :FOR J=1 TO 25 :NEXT J :BEEP0
1790  NEXT X
1800  IF MUL>0 THEN PRINT USING "## ﾊﾞｲ";MUL; ELSE PRINT
". ｽｶ !";
1810  BEEP
1820  M=M+CIN*MUL
1830 '
1840  FOR I=1 TO 1000 :NEXT I
1850  COLOR C7
1860  LOCATE 12,2 :PRINT "|      |"
1870  IF M>0 THEN 1120
1880 '
```



配当金の表示 INT PRINT USING

乱が抱えきれないほどの配当金をあてて大喜びしている。リストの2320～2440行は金額表示のサブルーチンだ。各コインが何枚あるかを示すようになっている。持ち金の変数はM1だ。

2350行で1000ビットコインの数を出している。持ち金を1000で割った答の整数部分がコインの数だね。こういう場合はINT関数を使うと便利だ。枚数が出たら2370行のPRINT USING文で表示すればいい。そして，次は100ビットコインの数の計算。この場合は，まず持ち金から1000ビットコインで表示した額だけ引いておいて，100で割る。つまり，1000の位を取りのぞいて，100の位の数字が何であるか調べているわけだ。以下のコインも同様のやり方で計算し表示している。

2390～2410行は，すでに表示しているコインと今回計算したコインの数を比べて表示の増減を判定しているところだ。1000ビットコインは表示の関係上15枚までにしているので，今回の結果が15枚以上の時は15に，15未満のときは現在のコインから増減させている。

2340～2420行は各コインを増やすサブルーチンにやるのか減らすサブルーチンにやるのか判定している。

2430行では，1000ビットコインが15枚以上だと，それを示すために“**”を表示するようにしている。これでお



金の表示はOKだ。

```
2320 REM ----------- ｵｶﾈ ﾋｮｳｼﾞ -----------
2330  M1=M
2340  FOR I=0 TO 3
2350   MN=INT(M1/MD(I))
2360   COLOR C7
2370   LOCATE 27+I*3,24 :PRINT USING "###";MN;
2380   M1=M1-MD(I)*MN
2390   IF MN>15 THEN MN=15
2400   IF MN>M(I) THEN GOSUB 2550
2410   IF MN<M(I) THEN GOSUB 2460
2420  NEXT I
2430  IF M(0)=15 THEN COLOR C6 :LOCATE 28,8 :PRINT "**";
2440  RETURN
```

2460～2530行はコインを減らすサブルーチン。

コインの減る数だけFOR～NEXT文で空白を表示してコインが減るように見せるわけだ。その間BEEP音も出して，ギャンブラーをくやしがらせているところはニクイではないか。

2550～2620行はコインを増やすサブルーチン。減らす時と同様なテクニックだが，空白の代わりに“■”を，色をコインごとに変えながら表示するようにしている。減らすサブルーチンの応用版だ。

```
2460 REM ----------- ｺｲﾝｦ ﾍﾗｽ -----------
2470  FOR J=M(I) TO MN+1 STEP -1
2480   LOCATE 28+I*3,23-J :PRINT "  ";
2490   BEEP1 :FOR K=1 TO 10 :NEXT K :BEEP0
2500  NEXT J
2510  M(I)=MN
2520 '
2530 RETURN
2540 '
2550 REM ----------- ｺｲﾝｦ ﾌﾔｽ -----------
2560  COLOR CM(I)
2570  FOR J=M(I)+1 TO MN
2580   LOCATE 28+I*3,23-J :PRINT "■";
2590   BEEP1 :FOR K=1 TO 10 :NEXT K:BEEP0
2600  NEXT J
2610  M(I)=MN
2620 RETURN
```



命令とその使い方 プログラムのテクニック

スロットボンバーに出てきた大事な命令をもう一度，整理して見てみよう。

CIRCLE：サークル／囲むという意味

《役目》円を描く。

PAINT：ペイント／ぬるという意味

《役目》指定した範囲をぬりつぶす。

LOCATE：ロケイト／場所を決めるという意味

《役目》表示の位置を指定する。

BEEP：ビープ／ビーという警笛を鳴らすという意味

《役目》一定時間だけ音を鳴らす。

BEEP 1, BEEP 0：ビープ／警笛を鳴らす意味

《役目》BEEP 1で音を鳴らし，BEEP 0で止める。

PRINT USING：プリント ユージング／～を使って表示するという意味

《役目》指定した書式で表示する。

RND：ランダマイズ／不規則なという意味の英語

《役目》0から1未満の乱数を発生する。

INT：インテジャー／整数という意味の英語

《役目》正の数は整数部分だけに，負の数は整数部分から1ひいたものに直す。

GET @，PUT @：ゲット・アット，プット・アット

《役目》GET @で表示を読み込み，PUT @で描き出す。



左のような命令や，基本的な命令を使って，1つの流れを持った動きをさせられる。命令をたくさん覚えるだけではプログラムを組めない。それを組み合わせるテクニックのバリエーションも同時に身につけていこう。

FOR～NEXTを使ったテクニック

①時間つぶし　FOR文とNEXT文のあいだに実行文を入れずにカラまわりさせる。

②データの読み込み　ループの中の実行文としてREAD文を入れ，データを配列に読み込む。ループの変数が配列の添字となる。

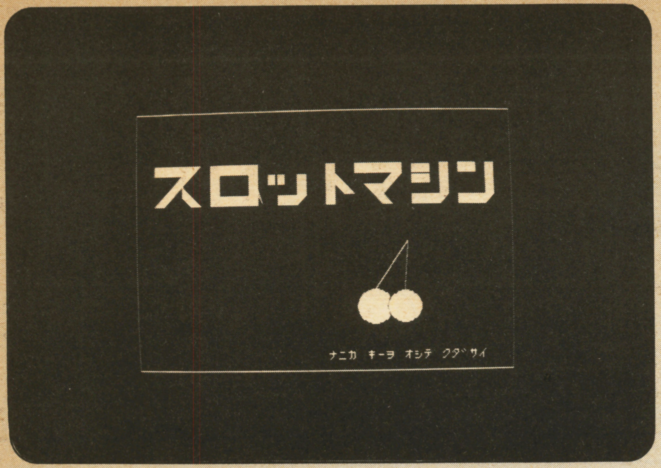
③ループの変数を使って仕事をさせる　ループを1回まわることで変数が増えるのを利用する。

表示を動かすテクニック

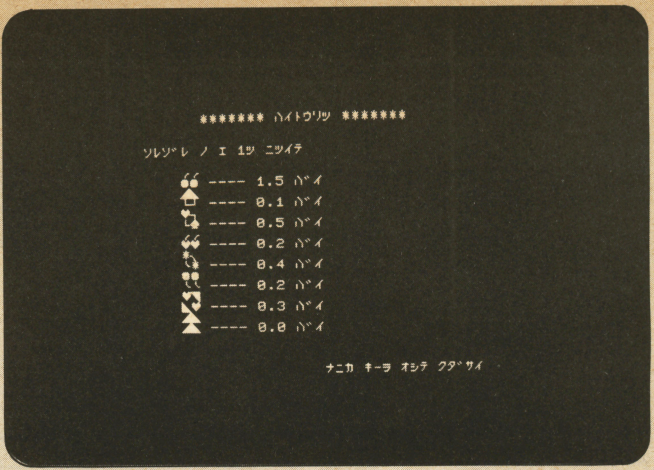
①LOCATE, PRINTを使う　表示したものの，動かしたい方向に新しいキャラクタをつけ加え，逆側を少し消す。これを，FOR～NEXTなどでくり返させれば，動いている感じがでる。

②GET @, PUT @を使う　表示したものを配列に読み込み，別の場所に描き出す。描き出す位置の指定のし方で，連続して動く感じもだせるね。

上のようなことを考えながら，もちろんフローチャートも合わせて，次ページからのリストを読んでみよう。細かいテクニックをどうつなげて，ゲームの全体を組みたてているのか，つながったリストを見るとよくわかる。



▲ゲームスタート



▲各絵柄の倍率はこの通り

スロットボンバー

●遊び方

一攫千金を夢見つつ，「スロットボンバー」ゲームに挑戦しよう。

はじめに，タイトルと，かわいらしいサクランボが表れる。画面の右下に「ナニカ　キーヲ　オシテクダサイ」と表示されたら，なんでもいい，キーを押そう。スペースバーが大きくて押しやすいね。

今度は，ねらうべき絵とそれぞれの倍率が表示される。どの絵で止めたらお金がもうかるか，よく考えよう。

次にキーを押すと，スロットマシンの本体が現れ，最初の資金がわたされる。持ち金500からスタートだ。かけ金を聞いてきたら，500の範囲で好きな額を入れよう。

スロットマシンの腕が縮んで伸びると，いよいよゲーム開始のブザーが鳴りひびく。3つ並んだうちの左側の窓が開いて，絵がカタコトカタコト……と回りだすぞ。

「ここだ！」と思うところでスペースバーを押そう。絵の回転がだんだんゆっくりになっていって，しまいにピタッと止まる。ひとつの絵が止まるとすぐに，まん中の絵がまわりはじめる。できるだけ同じ絵で止めるようにねらってみよう。倍率が高くなるぞ。2つ同じなら2倍に，3つなら3倍の倍率で計算される。

3つの絵を止め終わると，その回のトライアルの結果が表示される。最終的な倍率と，もうけを加算した新しい持ち金がしらされるんだ。

ここで持ち金が残っていれば，何回でも続けてトライアルできるぞ！

持ち金がゼロになったらゲームオーバー。「モウイチドヤリマスカ」と聞いてくる。さあ！　どうする？

《要注意》 ▲が3つ並ぶとキカイは大爆発。何もかもオジャンになってしまう。

(富士通)FM8 & FM7 全リスト

プログラムのタイトル

```
10 REM ****************************************
20 REM *                                      *
30 REM *            スロットマシン              *
40 REM *                                      *
50 REM *                  by  Nor  &  Lei     *
60 REM *                                      *
70 REM *                          Ver.1.0     *
80 REM ****************************************
90 REM
```

こうやって先頭にプログラム名を入れておくと後でわかりやすい。実行しよう。

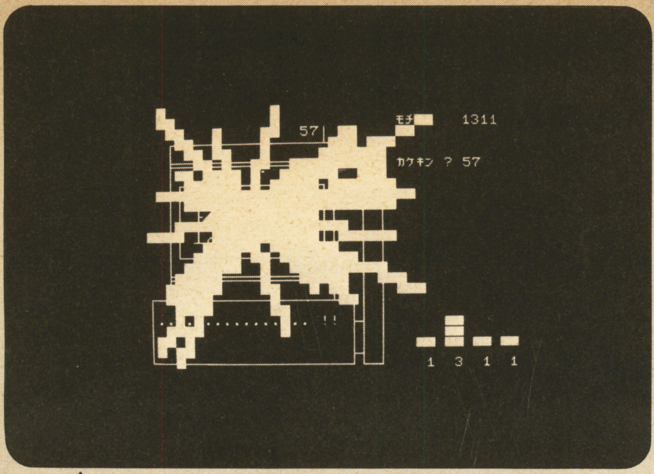
初期設定

```
100 REM ---------- ショキセッテイ ----------
110  COLOR ,0 :CLS
120  WIDTH 40,25 :CONSOLE 0,25,0,0
130  DIM H$(10),L$(10),CH(10),CL(10),G%(5)
140  DIM P(10),N(2),M(4),MD(4),CM(4)
150  M=500
160 '
```

持ち金500でスタート！

カラーデータの読み込み

```
170 REM ---------- カラー ヨミコミ ----------
180  READ C1,C2,C3,C4,C5,C6,C7
```



▲ ▲が3つそろうと……

変　数　表

A$	モウイチドヤリマスカの答え（'Y'か'N'）	M	持ち金
C1～C7	カラーコード	M( )	金種ごとのコインの枚数
CH( )	絵の上半分の色	MI	お金表示用の一時変数
CIN	入力金額(CIN$を数値に変換したもの)	MD( )	金種
CIN$	入力金額	MN	コインの枚数
CL( )	絵の下半分の色	MUL	倍率
CM( )	それぞれの金種の色	N	絵データの数
G%( )	GET@，PUT@用の配列	N(	どの絵でとまったか記憶用配列（左から0，1，2）
H$( )	絵の上半分の形	P( )	それぞれの絵の配当率
I，J，K，L	FOR～NEXTの変数	X	一時変数。表示のX座標指定に使用
L$( )	絵の下半分の形	Y	一時変数。表示のY座標指定に使用

```
190 '
200 REM ---------- カラー データ ----------
210  DATA 1,2,3,4,5,6,7
220 '
```

タイトルの表示

```
230 REM ---------- タイトル ----------
240  COLOR C7,C1 :CLS
250  LOCATE 0,5
260  PRINT "
270  PRINT "
280  PRINT "
290  PRINT "
```



タイトルを描く

```
300 '
310  LINE (0,0)-(639,199),PSET,C7,B
```

白ワクを描く

```
320 '
330 REM ---------- サクランボ ----------
340  CIRCLE(400,150),30,C2
350  PAINT (400,150),C2,C2
360  CIRCLE(450,150),30,C2
370  PAINT (450,150),C2,C2
380  CIRCLE(450,150),30,C1
390  LINE (455,100)-(400,137),PSET,C4
400  LINE (455,100)-(450,137),PSET,C4
```

サクランボを描く

```
410 '
420 REM ---------- オカネ カラー & キンガク リード ----------
430  FOR I=0 TO 3
440   READ MD(I),CM(I)
450  NEXT I
```

コインの色と金種を読み込む

```
460 '
```

配当率を表示する

```
470 REM ---------- データ リード ----------
480  N=0
490  READ CH(N)
500 '
510  READ H$(N),CL(N),L$(N),P(N),CH(N+1)
520  N=N+1
530  IF CH(N)<>0 THEN 510
540 '
```

回転する絵とその色，倍率のデータを読み込む

```
550  LOCATE 20,23 :PRINT "ナニカ キーヲ オシテ クダサイ";
560  IF INKEY$="" THEN 560
570 '
580 REM ---------- ハイトウリツ ヒョウジ -------------
590  COLOR C7,0 :CLS
600  PRINT "           ******* ハイトウリツ *******"
610  PRINT
620  PRINT
630  PRINT " ソレゾレ ノ エ 1ツ ニツイテ"
640  PRINT
650  FOR I=0 TO N-1
660   COLOR CH(I) :PRINT TAB(5);H$(I)
```

絵と倍率を表示する

```
670   COLOR CL(I) :PRINT TAB(5);L$(I);
680   COLOR C7
690   PRINT USING" ---- #.# ﾊﾞｲ";P(I)
700   BEEP
710   FOR J=1 TO 200:NEXT J
720  NEXT I
730 '
740  LOCATE 20,24 :PRINT "ﾅﾆｶ ｷｰｦ ｵｼﾃ ｸﾀﾞｻｲ";
750  IF INKEY$="" THEN 750
760 '
```

次のステップへ移るため！

キカイを描く

```
770 REM------- ｷｶｲ ﾋｮｳｼﾞ -------
780  CLS
790  PRINT :PRINT
800  PRINT"             |     |"
810  PRINT"   ┌─────────┴─────┴┐"
820  PRINT"   |               |"
830  PRINT"   ╞═══════════════╡"
840  PRINT"   |               |  ███"
850  PRINT"   |┌─────────────┐|  ███"
860  PRINT"   ||  ─────────  ||  ███"
870  PRINT"   || |██|██|██|  ||  ███"
880  PRINT"   || H██H██H██H  ||  | |"
890  PRINT"   || H██H██H██H  ||  | |"
900  PRINT"   || |██|██|██|  ||  | |"
910  PRINT"   ||  ─────────  ||  | |"
920  PRINT"   |└─────────────┘|  | |"
930  PRINT"   |               |  | |"
940  PRINT"   ╞═══════════════╡  | |"
950  PRINT"   |               |  | |"
960  PRINT" ┌─┴───────────────┴┐ | |"
970  PRINT"|                  || |"
980  PRINT"|                  H |"
990  PRINT"|                  || |"
1000  PRINT"|                  || |"
1010  PRINT"|                  H |"
1020  PRINT"└──────────────────┘└─┘";
1030 '
```

スロットマシン本体を表示する

```
1040 REM ---------- ｷｶｲﾆ ｲﾛｦ ﾂｸﾙ ----------
1050  PAINT (50, 30),C1,C7
1060  PAINT (50, 43),C2,C7
1070  PAINT (50, 47),C1,C7
1080  PAINT (50,131),C2,C7
1090  PAINT (50,135),C1,C7
1100  PAINT (65,150),C1,C7
1110 '
```

スロットマシンに色をつける

スタート！ かけ金は？

```
1120 REM ------- ｵｶﾈ ﾆｭｳﾘｮｸ -------
1130  GOSUB 2320
1140 '
1150  COLOR C7
1160  LOCATE 26,1 :PRINT USING "ﾓﾁｷﾝ ######";M
1170  LOCATE 26,5 :INPUT "ｶｹｷﾝ "; CIN$
1180  CIN=VAL(CIN$)
1190  IF CIN>M OR CIN<=0 THEN BEEP :GOTO 1170
1200  M=M-CIN
1210 '
1220  LOCATE 1,20 :PRINT "                              "
1230  LOCATE 26,1 :PRINT USING "ﾓﾁｷﾝ ######";M
1240  LOCATE 12,2 :PRINT USING "|#####|";CIN
1250  GOSUB 2320
1260 '
```

持ち金の表示

かけ金の入力を要求

かけ金が持ち金より大きければ，かけ金の再要求

持ち金，かけ金を数字とコインの数で表示する

マシンの始動

```
1270 REM ---------- ﾏﾄﾞﾆ ﾌﾀｦ ｽﾙ ----------
1280  COLOR C7
```

```
1290  FOR I=0 TO 2
1300   LOCATE 6+I*4, 9 :PRINT "██"
1310   LOCATE 6+I*4,10 :PRINT "██"
1320   LOCATE 6+I*4,11 :PRINT "██"
1330   LOCATE 6+I*4,12 :PRINT "██"
1340  NEXT I
1350 '
1360 REM ---------- ｳﾃﾞﾉﾊﾞｼ ----------
1370  FOR Y=4 TO 13
1380   LOCATE 22,Y   :PRINT "    "
1390   LOCATE 22,Y+4 :PRINT "███"
1400  NEXT Y
1410 '
1420  FOR I=1 TO 10
1430   BEEP1 :FOR J=1 TO 5 :NEXT J :BEEP0
1440  NEXT I
1450 '
1460  FOR Y=13 TO 4 STEP -1
1470   LOCATE 22,Y+4 :PRINT "| |"
1480   LOCATE 22,Y   :PRINT "███"
1490  NEXT Y
1500 '
```

スロットマシンの窓が閉じる (1290–1340)

スロットマシンの腕が縮んで伸びる (1370–1490)

絵を止めろ!

```
1510 REM ------- ｳｺﾞｶｽ ! -------
1520  FOR I=0 TO 2
1530   J=INT(RND(1)*N)
1540   IF INKEY$="" THEN GOSUB 2160 :GOTO 1540
1550   FOR K=1 TO INT(RND(1)*N+3)
1560    GOSUB 2160
1570    FOR L=1 TO K*15 :NEXT L
1580   NEXT K
1590   N(I)=J
1600  NEXT I
1610 '
```

回転がはじまり、キー入力で止まる (1520–1600)

お金の計算

```
1620 REM ------- ｹｲｻﾝ -------
1630  MUL=0
1640  FOR I=0 TO 2
1650   MUL=MUL+P(N(I))
1660  NEXT I
1670 '
1680  IF MUL=0 THEN 2640
1690 '
1700  IF N(0)=N(1) THEN MUL=MUL+P(N(0))*2
1710  IF N(0)=N(2) THEN MUL=MUL+P(N(0))*2
1720  IF N(1)=N(2) THEN MUL=MUL+P(N(1))*2
1730  MUL=INT(MUL)
1740 '
1750  COLOR C6
1760  FOR X=1 TO 13
1770   LOCATE X,20 :PRINT ".";
1780   BEEP1 :FOR J=1 TO 25 :NEXT J :BEEP0
1790  NEXT X
1800  IF MUL>0 THEN PRINT USING "## ﾊﾞｲ";MUL; ELSE PRINT ". ｽｶ"
1810  BEEP
1820  M=M+CIN*MUL
1830 '
1840  FOR I=1 TO 1000 :NEXT I
1850  COLOR C7
1860  LOCATE 12,2 :PRINT "|      |"
1870  IF M>0 THEN 1120
1880 '
```

3つの絵の倍率を合計する (1630–1660)

▲▲▲ で爆発のサブルーチンへ (1680)

同じ絵が出れば倍率加算 (1700–1730)

倍率の表示 (1750–1810)

持ち金を増やす計算 (1820)

持ち金が残っていればもう一度トライアル! (1840–1870)

エンディング

```
1890 REM ------- ｵｶﾈｶﾞﾅｸﾃ ｵﾜﾘ -------
1900  COLOR C2
```

```
1910  FOR I=0 TO 2
1920   LOCATE 6+I*4, 9 :PRINT "  "
1930   LOCATE 6+I*4,10 :PRINT "\/"
1940   LOCATE 6+I*4,11 :PRINT "/\"
1950   LOCATE 6+I*4,12 :PRINT "  "
```

窓の中を消していく

```
1960 '
1970   FOR J=1 TO 10
1980    BEEP1 :FOR K=1 TO 10 :NEXT K :BEEP0
1990   NEXT J
2000  NEXT I
2010 '
2020   PAINT (45,30),0,C6
2030   FOR J=1 TO 150
2040    BEEP1 :FOR K=1 TO 10 :NEXT K :BEEP0
2050   NEXT J
```

スロットマシンを消す

```
2060 '
2070  COLOR C7
2080  LOCATE 1,20 :PRINT "ﾓｳｲﾁﾄﾞ ﾔﾘﾏｽｶ ? (Y/N)";
2090  A$=INKEY$ :IF A$="" THEN 2090
2100  IF A$="Y" OR A$="y" THEN M=500 :GOTO 580
```

もう一度やるなら配当率を再表示

```
2110  IF A$<>"N" AND A$<>"n" THEN BEEP :GOTO 2090
2120 '
2130  CLS
2140  END
```

画面を消して終わり

```
2150 '
```

回転のサブルーチン

```
2160 REM ------- ｶｲﾃﾝ -------
2170  J=(J+1) MOD N
2180  BEEP1 :BEEP0
2190  GET@A(6+I*4,10)-(7+I*4,12),G%
2200  PUT@A(6+I*4, 9)-(7+I*4,11),G%
2210 '
2220  LOCATE 6+I*4,12
2230  COLOR CL(J) :PRINT L$(J)
2240 '
2250  GET@A(6+I*4,10)-(7+I*4,12),G%
2260  PUT@A(6+I*4, 9)-(7+I*4,11),G%
2270 '
2280  COLOR CH( (J+1) MOD N)
2290  LOCATE 6+I*4,12 :PRINT H$((J+1) MOD N)
2300  RETURN
```

表示を少しずつ上にずらし，下に新しい絵を加えることで，回転する感じをだしている

```
2310 '
```

コインを表示するサブルーチン

```
2320 REM ---------- ｵｶﾈ ﾋｮｳｼﾞ ----------
2330  M1=M
2340  FOR I=0 TO 3
```

3種類の金種

```
2350   MN=INT(M1/MD(I))
2360   COLOR C7
2370   LOCATE 27+I*3,24 :PRINT USING "###";MN;
2380   M1=M1-MD(I)*MN
```

金種ごとのコインの数を計算

```
2390   IF MN>15 THEN MN=15
```

コインが15より多い時はすべて15とみなす

```
2400   IF MN>M(I) THEN GOSUB 2550
2410   IF MN<M(I) THEN GOSUB 2460
```

前の持ち金と現在の持ち金を比べて，それぞれのサブルーチンへ

```
2420  NEXT I
2430  IF M(0)=15 THEN COLOR C6 :LOCATE 28,8 :PRINT "**";
```

15より多いコインの上に印をつける

```
2440  RETURN
2450 '
```

コインを減らすサブルーチン

```
2460 REM ---------- ｺｲﾝｦ ﾍﾗｽ ----------
2470  FOR J=M(I) TO MN+1 STEP -1
2480   LOCATE 28+I*3,23-J :PRINT "  ";
2490   BEEP1 :FOR K=1 TO 10 :NEXT K :BEEP0
2500  NEXT J
2510  M(I)=MN
```

表示されているコインの上に重ねてスペースを描いて，コインを減らす

```
2520 '
```

```
2530 RETURN
2540 '
```

コインを増やすサブルーチン

```
2550 REM ---------- ｺｲﾝｦ ﾌﾔｽ ----------
2560  COLOR CM(I)
2570  FOR J=M(I)+1 TO MN
2580   LOCATE 28+I*3,23-J :PRINT "■■";
2590   BEEP1 :FOR K=1 TO 10 :NEXT K:BEEP0
2600  NEXT J
2610  M(I)=MN
2620 RETURN
2630 '
```

コインを新しく描きたす（2560〜2620）

▲が3つでバクハツ

```
2640 REM ---------- ﾊﾞｸﾊﾂ ----------
2650  COLOR C6
2660  FOR X=1 TO 16
2670   LOCATE X,20 :PRINT ".";
2680   BEEP1 :FOR J=1 TO 25 :NEXT J :BEEP0
2690  NEXT X
2700  PRINT " !!"
2710  BEEP
2720 '
2730  FOR I=1 TO 1000 :NEXT I
2740 '
2750  COLOR C2
2760  FOR I=0 TO 50
2770   X=INT(RND(1)*30) :Y=INT(RND(1)*25)
2780   LINE (10,10)-(X,Y),"■",C2
2790   BEEP
2800  NEXT I
2810 '
2820  FOR I=1 TO 50
2830   BEEP1 :FOR J=1 TO 10 :NEXT J :BEEP0
2840  NEXT I
2850 '
2860  LOCATE 0,24
2870  FOR I=0 TO 24 :PRINT :NEXT I
2880 '
2890  COLOR 7
2900  CLS
2910  END
2920 '
2930 REM ---------- ｷﾝｶﾞｸ & ｲﾛ ﾃﾞｰﾀ ----------
2940  DATA 1000,1, 100,2, 10,3, 1,4
2950 '
2960 REM ---------- ﾃﾞｰﾀ ----------
2970  DATA 4,◢◢
2980  DATA 2,●●,1.5
2990  DATA 4,▲
3000  DATA 5,└┘, .1
3010  DATA 2,♥┐
3020  DATA 1,└♠, .5
3030  DATA 4,◢◢
3040  DATA 2,♥♥, .2
3050  DATA 6,*┐
3060  DATA 6,└*, .4
3070  DATA 5,●●
3080  DATA 3,◣◣, .2
3090  DATA 2,♥◥
3100  DATA 2,◣♥, .3
3110  DATA 5,▲
3120  DATA 5,▲,  0
3130  DATA 0
```

- 画面に「……!!」と表示する（2650〜2710）
- 時間つぶしのループ（2730）
- 50ヶの■を，不規則に並べて，爆発の感じをだしている（2750〜2800）
- 画面を消す（2860〜2870）
- 終わり（2890〜2910）
- コインの金種と色のデータ（2940）
- 回転する絵の色，形，倍率のデータ（2970〜3130）

(NEC) PC8001への移植のPoint (PC8801でも動きます)

CONSOLE w, x, y, z 第4パラメータzのカラーモードを PC は1に設定しよう。

画面構成(ドット数) PC は160 ×100 (WIDTH 80の時) か，80×100 (WIDTH40の時) なので，それぞれに応じて換算する。

CLSはPRINT CHR$(12) で表す。

COLOR x, y PC は第2パラメータyはあるが意味が違うので背景色は指定できない。

CIRCLE PC にはないのでキャラクタを組み合わせて表現する。

PAINT PC にはない。■のキャラクタで色をつけることもできるが，ここでは色をつけないままにする。

```
110  PRINT CHR$(12)
120  WIDTH 40,25 :CONSOLE 0,25,0,1

240  COLOR C7 :PRINT CHR$(12)

310  LINE (0,0)-(79,99),PSET,C7,B

340  COLOR C4
345  LOCATE 25,12 :PRINT "      /1 "
350  LOCATE 25,13 :PRINT "     /| "
```

```
355  LOCATE 25,14 :PRINT "    / | "
360  LOCATE 25,15 :PRINT "   /  | "
365  LOCATE 25,16 :PRINT "  /   | "
370  COLOR C2
375  LOCATE 25,17 :PRINT "■■■■"
380  LOCATE 25,18 :PRINT "■■■■"
385  LOCATE 25,19 :PRINT "■■■■"
390  LOCATE 25,20 :PRINT "■■■■"

780  PRINT CHR$(12);
```

1050行から1110行までは削除

```
1165  LOCATE 26,5 :PRINT "ｶｹｷﾝ     "
1170  LOCATE 31,5 :INPUT CIN$
```

2020行は削除

(日立) LEVEL 3への移植のPoint

CONSOLE w, x, y, z 第4パラメータなし。いつでもカラーモード。

CIRCLE L3 はなし。FOR～NEXTと三角関数で座標を決め，LINEで線をひく。

PAINT L3 のPAINTはドット以外を境界とみなさない。

GET@, PUT@ L3 はなし。LOCATEを使う。

BEEP1, BEEP0 一定時間だけ音を出すBEEPしかない。

WIDTH x, y 第2パラメータなし。25行で固定。

モード SCREENの第1パラメータでドットのこまかいハイレゾリューションモードを，第3パラメータでキャラクタを表示するノンインターレースモードを指定する。第2パラメータはここでは特に指定の必要はない。

```
105  SCREEN 1,,1

120  WIDTH 40 :CONSOLE 0,25,0

341  X=400 :Y=150
342  X1=40 :Y1=0
343  FOR I=0 TO 3.142*2 STEP .2
344   X2=40*COS(I) :Y2=15*SIN(I)
345   LINE (X+X1,Y+Y1)-(X+X2,Y+Y2),
     PSET,C2
346   X1=X2 :Y1=Y2
347  NEXT I

361  X=450 :Y=150
362  X1=40 :Y1=0
363  FOR I=0 TO 3.142*2 STEP .2
364   X2=40*COS(I) :Y2=15*SIN(I)
365   LINE (X+X1,Y+Y1)-(X+X2,Y+Y2),
     PSET,C2
366   X1=X2 :Y1=Y2
367  NEXT I
```

380行は削除

```
390  LINE (455,90)-(400,137),PSET,C4
400  LINE (455,90)-(450,137),PSET,C4
```

1050行から1110行までは削除

```
1170  LOCATE 26,5 :PRINT "ｶｹｷﾝ     "
1175  LOCATE 31,5 :INPUT CIN$

1420  FOR I=1 TO 5
1430   BEEP
1440  NEXT I

1780  BEEP

1970 '
1980   BEEP
1990 '
```

2020行は削除

```
2040   BEEP

2170  BEEP

2190  COLOR CH( (J+2) MOD N)
2200  LOCATE 6+I*4,12 :PRINT H$( (J+
      2) MOD N)
2210  COLOR CL( (J+1) MOD N)
2220  LOCATE 6+I*4,11 :PRINT L$( (J+
      1) MOD N)
2230  COLOR CH( (J+1) MOD N)
2240  LOCATE 6+I*4,10 :PRINT H$( (J+
      1) MOD N)
2250  COLOR CL(J)
2260  LOCATE 6+I*4, 9 :PRINT L$(J)
2270 '
2280  J=(J+1) MOD N

2490   BEEP

2590   BEEP

2680   BEEP

2820  FOR I=1 TO 10
2830   BEEP
```

(シャープ) MZ700への移植のPoint

CONSOLE w, x, y, z MZ はカラーモードで固定されているので，第3，第4パラメーターとも省略する。(MZ の第3パラメータはスクロールの開始位置を，第4パラメータはスクロール字数の設定である)

画面構成(ドット数) 80×50だから換算が必要。

CIRCLE MZ はなし。キャラクタで表現する。

PAINT MZ はなし。キャラクタで表現する。

GET@, PUT@ MZ はなし。LOCATEを使う。

LINE MZ なし。PSETを使う。

WIDTH x, y MZ はなし。40字×25行で固定。

INKEY$ プログラムには関係ないが，入力する時に MZ では小文字が入力できない。

★これらの機種で遊ぶにはFM8用のものを

```
120  CONSOLE 0,25,0,40
130  DIM H$(10),L$(10),CH(10),CL(10)

310   FOR X=0 TO 79
312    PSET(X,0,C7) :PSET (X,49,C7)
314   NEXT X
316   FOR Y=0 TO 49
318    PSET(0,Y,C7) :PSET(79,Y,C7)
320   NEXT Y

335  COLOR C4
340  LOCATE 25,12 :PRINT "       /|"
345  LOCATE 25,13 :PRINT "      / |"
350  LOCATE 25,14 :PRINT "     /  |"
355  LOCATE 25,15 :PRINT "    /   |"
360  LOCATE 25,16 :PRINT "   /    |"
365  COLOR C2
370  LOCATE 25,17 :PRINT "■■ ■■"
375  LOCATE 25,18 :PRINT "■■ ■■"
380  LOCATE 25,19 :PRINT "■■ ■■"
385  LOCATE 25,20 :PRINT "■■ ■■"

545  COLOR C7

740  LOCATE 20,23 :PRINT "ﾅﾆｶ ｷｰｦ ｵｼﾃ
     ｸﾀﾞｻｲ";
```

1050行から1110行までは削除

```
1170  LOCATE 26,5 :PRINT "ｶｹｷﾝ       "
1175  LOCATE 31,5 :INPUT CIN$

1430  BEEP1 :FOR J=1 TO 5 :NEXT J:BEEP0

1980    BEEP1 :FOR K=1 TO 10 :NEXT K:BE
      EP0
```

2020行は削除

```
2040   BEEP1 :FOR K=1 TO 10 :NEXT K:BEE
      P0

2100  IF A$="Y" THEN M=500 :GOTO 580
2110  IF A$<>"N" THEN BEEP :GOTO 2090

2170  BEEP1 :BEEP0
2180 '
2190  COLOR CH( (J+2) MOD N)
2200  LOCATE 6+I*4,12 :PRINT H$( (J+2)
      MOD N)
2210  COLOR CL( (J+1) MOD N)
2220  LOCATE 6+I*4,11 :PRINT L$( (J+1)
      MOD N)
2230  COLOR CH( (J+1) MOD N)
2240  LOCATE 6+I*4,10 :PRINT H$( (J+1)
      MOD N)
2250  COLOR CL(J)
2260  LOCATE 6+I*4, 9 :PRINT L$(J)
2270 '
2280  J=(J+1) MOD N
2290  RETURN
2300 '
2310 '

2490   BEEP1 :FOR K=1 TO 10 :NEXT K:BE
      EP0

2590   BEEP1 :FOR K=1 TO 10 :NEXT K:BE
      EP0

2780   LOCATE X,Y :PRINT "■";

2830   BEEP1 :FOR J=1 TO 20 :NEXT J:BE
      EP0
```

(東芝)PASOPIAへの移植のPoint

CONSOLE w, x, y, z PASにはなし。

画面構成(ドット数) 288×192に換算が必要。

COLOR PASは1行単位で実行される。1行の中で色を変えたい時は，カラー・アトリビュート・キャラクタを使う。このキャラクタはCHR$関数で出力した場所から右側が指定の色になる。

PAINT ドット以外を境界とみなさないので，キャラクタで描いた中には色をつけられない。

GET@, PUT@ PASではディスクベーシックでは，この命令はあるが，機能が違うのでここではLOCATEに変更してある。

BEEP1, BEEP0 PASではSOUND x, yがこれにあたる。xは音の高さを，yは音の長さを指定。

モード ドットのこまかいファイングラフィックモードをSCREEN 2で指定する。

WIDTH x, y 第2パラメータなし，25行で固定。第1パラメータは1から80まで指定できる。このゲームでは36文字を指定。

LINE キャラクタで線をひけないので，直線をひく時のPSETが不要になる。

```
115  SCREEN 2
120  WIDTH 36
130  DIM H$(10),L$(10),CH(10),CL(10)
140  DIM P(10),N(2),C$(7),M(3),MD(3),
     CM(3)
170 REM ---------- ｶﾗｰ ﾖﾐｺﾐ ---------
175  READ C1,C2,C3,C4,C5,C6,C7
180  FOR I=1 TO 7
185   READ C : C$(I)=CHR$(C)
190  NEXT I
215 DATA 249,250,251,252,253,254,255
310  LINE (0,0)-(287,191),C7,B
340  CIRCLE(180,150),15,C2 ┐
350  PAINT (180,150),C2,C2 │
360  CIRCLE(200,150),15,C2 │
370  PAINT (200,150),C2,C2 ┘
380  CIRCLE(200,150),15,C1
390  LINE (205,100)-(180,137),C4
400  LINE (205,100)-(200,137),C4
```

(340～370) ROMベーシックではPAINTの第2パラメータC2をとる

```
550  LOCATE 15,22 :PRINT C$(7);"ﾅﾆｶ
     ｷｰｦ ｵｼﾃ ｸﾀﾞｻｲ";
670   COLOR CL(I) :PRINT TAB(5);L$(I)
     ;C$(7);
700   SOUND 45,15
740  LOCATE 15,23 :PRINT C$(7);"ﾅﾆｶ
     ｷｰｦ ｵｼﾃ ｸﾀﾞｻｲ";
790  PRINT
```

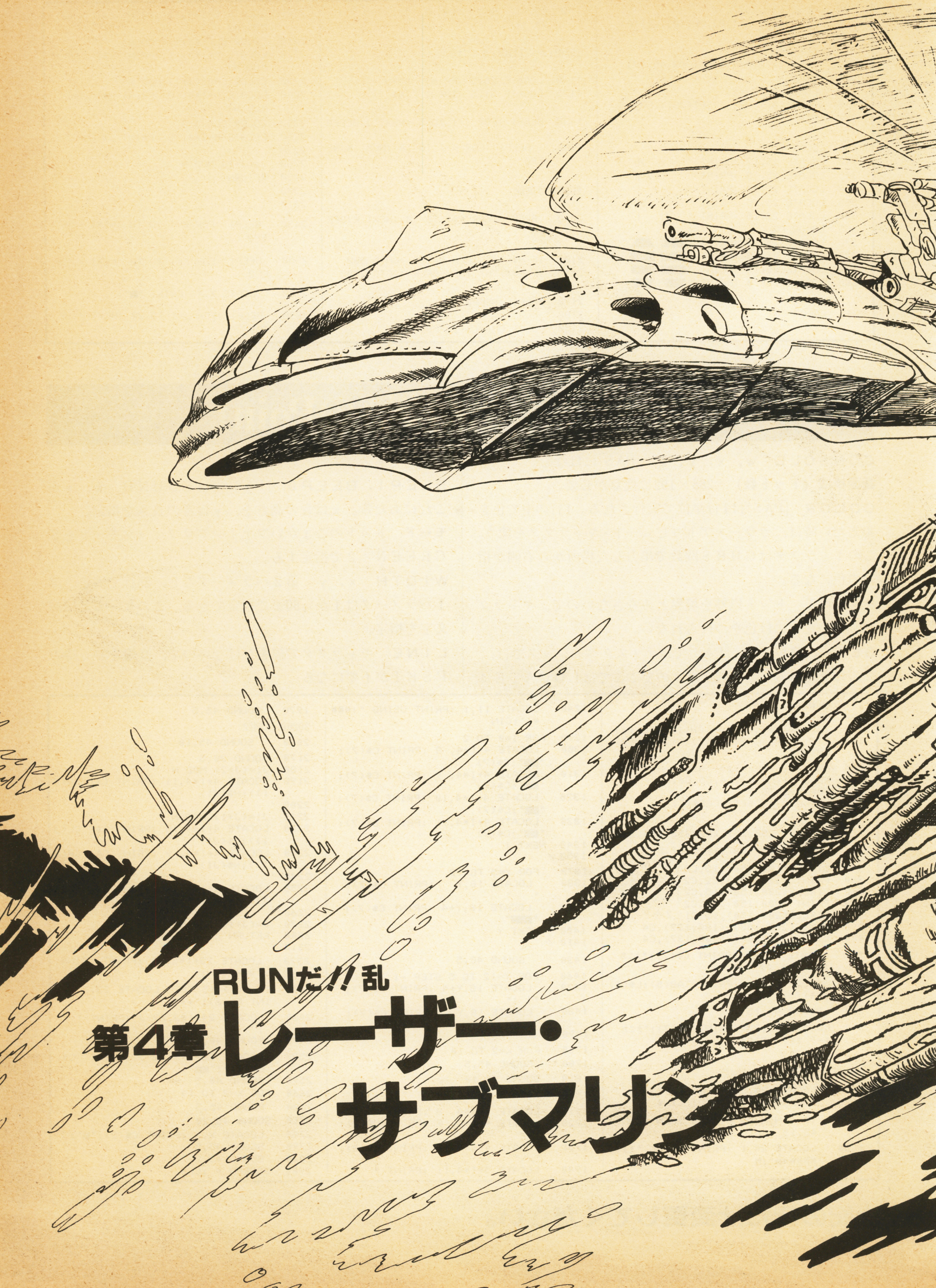
1050行から1100行までは削除

```
1170   LOCATE 24,5 :PRINT "ｶｹｷﾝ     "
1175   LOCATE 29,5 :INPUT CIN$
1190   IF CIN>M OR CIN<=0 THEN SOUND
       45,15 :GOTO 1170
1230   LOCATE 24,1 :PRINT USING "ﾓﾁｷﾝ
       ######";M
1240   LOCATE 11,1 :PRINT USING "|###
       #|";CIN
1290  FOR I=0 TO 2
1300   LOCATE 4+I*4, 8 :PRINT C$(7);
      "■■";C$(7)
1310   LOCATE 4+I*4, 9 :PRINT C$(7);
      "■■";C$(7)
1320   LOCATE 4+I*4,10 :PRINT C$(7);
      "■■";C$(7)
1330   LOCATE 4+I*4,11 :PRINT C$(7);
      "■■";C$(7)
1340  NEXT I
1350 '
1370   FOR Y=5 TO 12
1380    LOCATE 18,Y   :PRINT C$(7);
      "   "
1390    LOCATE 18,Y+4 :PRINT C$(7);
      "■■■"
1400   NEXT Y
1410 '
1430     SOUND 40,5
1460   FOR Y=12 TO 5 STEP -1
1470    LOCATE 18,Y+4 :PRINT C$(7);
      "| |"
1480    LOCATE 18,Y   :PRINT C$(7);
      "■■■"
1490   NEXT Y
1760   FOR X=2 TO 10
1780    SOUND 45,5
1810   SOUND 45,10
1860   LOCATE 11,1 :PRINT "|    |"
1910   LOCATE 5+I*4, 8 :PRINT "  "
1920   LOCATE 4+I*4, 9 :PRINT C$(2);
      "\/"
1930   LOCATE 4+I*4,10 :PRINT C$(2);
      "/\"
1940   LOCATE 5+I*4,11 :PRINT "  "
1970    SOUND 45,5
```

2020行は削除

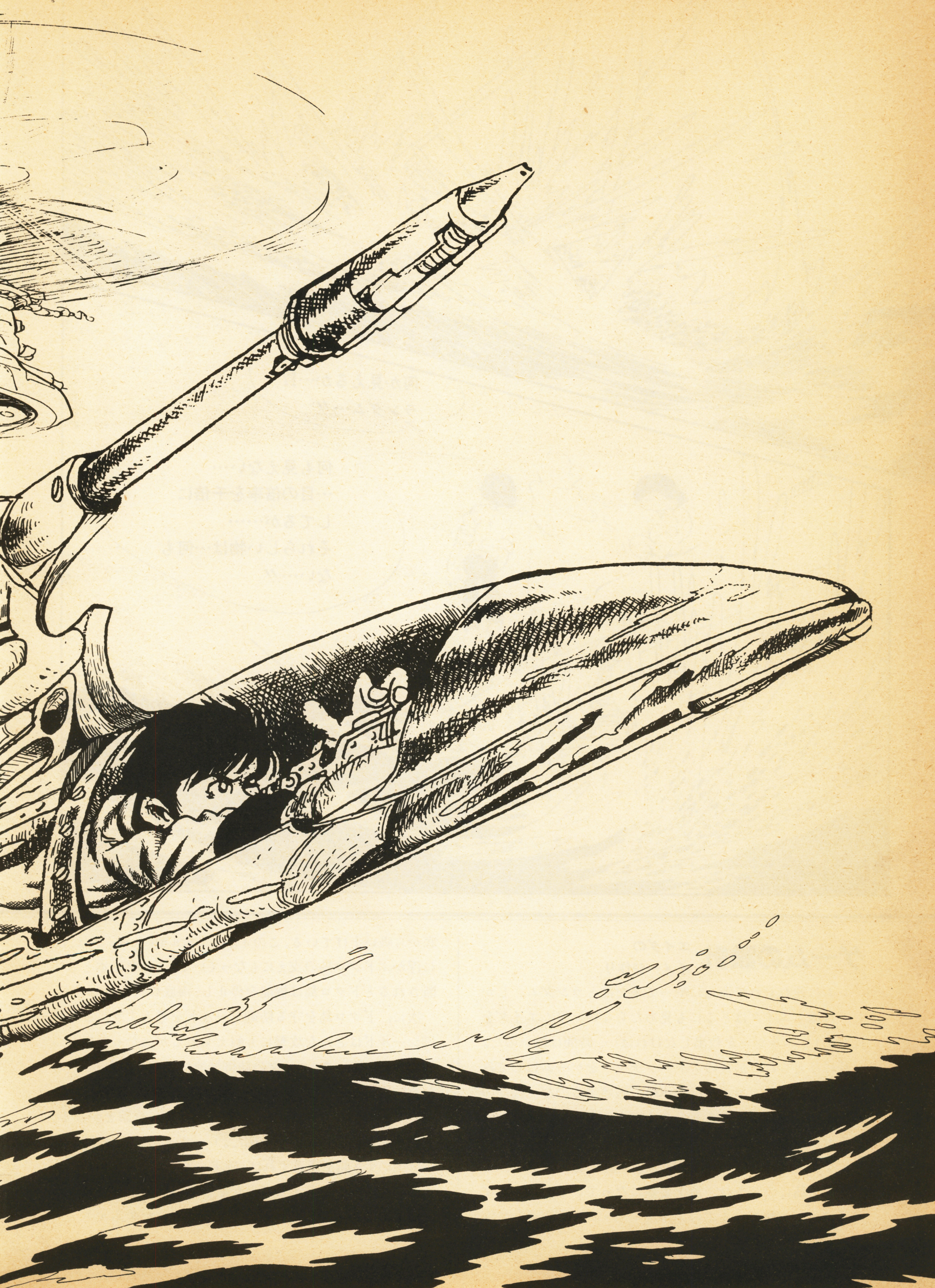
```
2040    SOUND 45,10
2170  SOUND 30,5
2180  J1=(J+2) MOD N
2190  LOCATE 4+I*4,11 :PRINT C$( CH(J
      1));H$(J1)
2200 '
2210  J1=(J+1) MOD N
2220  LOCATE 4+I*4,10 :PRINT C$( CL(J
      1));L$(J1)
2230  LOCATE 4+I*4, 9 :PRINT C$( CH(J
      1));H$(J1)
2240 '
2250  LOCATE 4+I*4, 8 :PRINT C$( CL(J
      )); L$(J)
2260 '
2270  J=(J+1) MOD N
2300  RETURN
2310 '
2370   LOCATE 23+I*3,23 :PRINT USING
       "###";MN;
2430  IF M(0)=15 THEN LOCATE 23,7 :PR
      INT C$(6);"**";
2480   LOCATE 24+I*3,22-J :PRINT "  "
       ;
2490   SOUND 40,5
2580   LOCATE 23+I*3,22-J :PRINT  C$(
       CM(I));"■■";
2590   SOUND 40,5
2660  FOR X=2 TO 14
2680   SOUND 45,5
2710  SOUND 45,20
2780   LOCATE X,Y :PRINT C$(2);"■";
2790   SOUND 45,5
2830   SOUND 45,2
2900  SCREEN 0 :CLS
```

上記の行だけ変更してください。



RUNだ!!乱

第4章 レーザー・サブマリン





ゲームの設計 エイダのイメージ作り

レムに連れられて海辺にやって来た乱とランダム。しかし，目指すエイダはどこにも見あたらない。早くエイダを探し出し，エルシーを救出しなければ…。時間は無いのだ。

乱に与えられた最後のアドベンチャーをゲームにしてみよう。海に浮かぶエイダを潜水艇で破壊するのだ。物語のクライマックスでもあるし，今までよりも少しこったプログラムにしてみたい。

まずエイダのイメージを作ろう。エイダは，ドクロ島という人工島に偽装され，海上要塞になっている。エネルギー源はマグマ熱だ。４本のエネルギー・パイプを通して海底から吸い上げている。この４本のエネルギー・パイプは，海に浮かぶドクロ島の支えにもなっている。これを４本とも切断されるとドクロ島は機能を停止し，海底に沈んでしまう。

次に，ドクロ島を守る防御シールドを考えよう。この防御シールドはどんな攻撃にも耐えられるが，たったひとつ弱点を持っている。エイダの防御機能にはバグがあり，そのため防御シールドにわずかなすき間が生じるのだ。潜水艇は，このすき間をねらってエネルギー・パイプに攻撃をかける。

潜水艇の武器はレーザー砲だ。レーザーがみごとに防御シールドのすき間を通過すれば，確実にエイダのエネルギー・パイプにダメージを与えることができる。だが，ドクロ島は自己修復機能が充実しているので，破壊されたエネルギー・

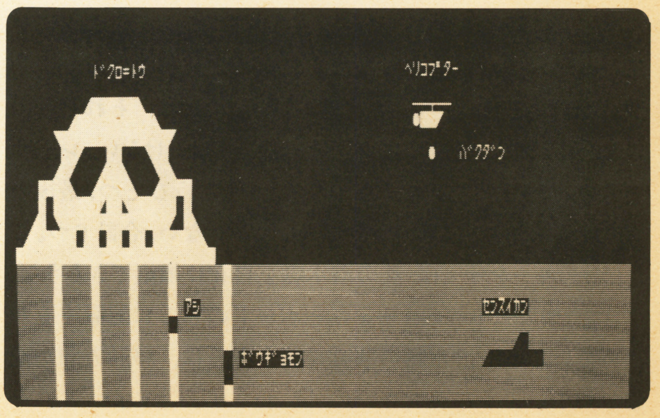


パイプも時間がたつと修理されてしまう。うかうかしていると，切断したエネルギー・パイプも修復されてしまうわけだ。

ドクロ島の周辺海域には，警戒用の対潜ヘリを配置することにしよう。対潜ヘリは潜水艇に対して空から爆雷の雨を降らせるが，潜水艇は対潜ヘリを攻撃できない。乱は，次々に落ちてくる爆雷をよけながらレーザーを発射しなければならないわけだ。そこで，どうしてもよけきれなかったときのために潜水艇はワープを1回だけ使えるようにしてある。

最後に，エルシーを救出するまでのタイム・リミットを設ければおもしろくなる。

ゲームのイメージがだいたい頭に浮かんだところで，ドクロ島，潜水艇，対潜ヘリのディスプレイ上の配置を考えよう。これをしっかり決めておくとプログラム作りが楽になる。今回のプログラムでは，配置を下の写真のようにしてみた。





潜水艇の操縦方法 テンキーを使う

ゲームの設計が終わったら，遊び方を考えよう。潜水艇をキーボードから操縦するにはどんな命令が必要だろうか？潜水艇を上・下・左・右の4方向に動かすには，それぞれの方向を指令する4つのキーを決めなくてはいけない。レーザーの発射を指令するキーも1つほしい。もうひとつ，最後の切り札であるワープの指令キーも欠かせない。合計6つ。これだけのキーを使えば，乱の潜水艇を自在に操縦できそうだ。

さて，実際に6つのキーをキーボードのどこに配置するか決めていこう。潜水艇の移動に使うキーは，移動方向が連想しやすいようにテンキーの「2」・「4」・「6」・「8」を使う。レーザー砲はゲーム中によく使うので大きくて押しやすいスペース・バーだ。最後のワープのキーは何回も使うものではないので「.」(ピリオド・キー）くらいにしておけばいいだろう。決まったキーを下にまとめておく。

ゲームで使うキー

	8 UP	
LEFT 4	潜水艇	6 RIGHT
	2 DOWN	

レーザー砲発射　SPACE BAR

. ワープ



ゲームの説明はリストの940〜960行だ。最初にGOSUB 3940となっているが，そこにはFOR〜NEXT文しかない。間に何も入っていないFOR〜NEXT文を使って時間をつぶしているわけだ。960行では，文字色の指定（白のカラーコード*は7）と画面クリアを行い，3300行に飛ばしている。

実際のゲームの説明は3300〜3920行で処理している。つまり，3300行以降がゲーム説明用のサブルーチン*だ。ここではゲームの説明がPRINT文で書き出してある。COLOR文やLOCATE文で見やすいように色つけやレイアウトをしてあるが，プログラマーの美的センスが問われるところだ。

```
3300 REM ----- アソビカタ ノ セツメイ -----
3310  BEEP :LOCATE 20,8 :PRINT "アソビカタ ノ セツメイ ガ
      ヒツヨウ デスカ (Y/N) ?";
3320  GOSUB 3250 :ANS$=I$ :PRINT ANS$
3330 IF ( ANS$<>"Y" ) AND ( ANS$<>"y" ) THEN RETURN
3340  CLS :COLOR C5 :LOCATE 25,2 :PRINT "***"
3350  LOCATE 45,2 :PRINT "***"
3360  COLOR C6 :LOCATE 30,2 :PRINT "アソビカタ ノ セツメイ";
3370 '
3380  COLOR C7 :LOCATE 14,6 :PRINT "コノ ゲーム ハ, キメラレタ
      ジカン ナイ ニ";
3390  COLOR C2 :PRINT " 'ドクロ=キチ'";
3400  COLOR C7 :PRINT " ヲ ハカイ スル ゲーム デス。"
3410  GOSUB 3940
3420 '
3430  LOCATE 7,9 :PRINT "アナタ ハ, "; :COLOR C5 :PRINT
      "'クロ ノ センスイカン' ";
3440  COLOR C7 :PRINT "ヲ ツカイ, "; :COLOR C6 :PRINT
      "'ビーム ホウ' ";
3450  COLOR C7 :PRINT "デ "; :COLOR C2
      :PRINT "'ドクロ=トウ' ノ アシ ";
3460  COLOR C7 :PRINT "ヲ コウゲキ シマス。"
3470  GOSUB 3940
3480 '
3490  LOCATE 7,11 :COLOR C2 :PRINT "'ドクロ=トウ' ノ アシ";
3500  COLOR C7 :PRINT " ヲ, 4 ホン トモ ";
3510  COLOR C6 :PRINT "'ビーム ホウ'"; :COLOR C7 :PRINT
     " デ ハカイスルト, ";
3520  COLOR C3 :PRINT "アナタ ノ カチ "; :COLOR C7 :PRINT "デス。"
3530  GOSUB 3940
```

＊ カラーコード：それぞれの色に割り当てられた番号。1対1で対応している。

＊ サブルーチン：1つのプログラムで何回も使うまとまった命令。GOSUB文で呼び出す。



フローチャート　プログラムのブロックの考え方

ゲームのイメージと遊び方を決めたら，今度は実際にプログラムを作成する。でも，思いついたままプログラムリストをいきなりマイコンに入れていくと，うまくいかないことが多い。ゲームのイメージを部分（これをブロックという）に分けて，ブロックごとにまとめると作りやすい。そこで，このゲームにはどんなブロックが必要か考えてみよう。

プログラムの最初で必要な処理は次の5つだ。

1．配列の宣言や，WIDTH 文・DEFINT 文・CONSOLE 文など，画面の状態を設定
2．ゲームのタイトルをディスプレイに表示
3．各変数に最初に必要な値を代入
4．潜水艇，対潜ヘリ，などのゲームで使う絵を定義
5．ゲームの遊び方の説明をディスプレイに表示

1と3のような，配列の宣言や変数の初期値の決定などの仕事を合わせて“初期化”(INITIALIZE／イニシャライズ）と呼んでいる。5の遊び方の説明は前のページで説明したね。これをわかりやすくやっておくのはゲームプログラマーのマナーだ。しっかりやっておかないと，遊ぶ人にとても不親切なゲームになってしまう。こうして1項目ずつブロックにして作り，最後にそれらをまとめてひとつのプログラムにまとめれば作りやすいわけだ。こうすれば，何人かで手分けしてプログラムを作ることもできる。

＊ フローチャート：考え方の流れをわかりやすく図に示したもの。



ゲームの用意が終わったら戦闘の部分を考えよう。大きく分けて以下の３つを作ればよさそうだ。

1．潜水艇の処理

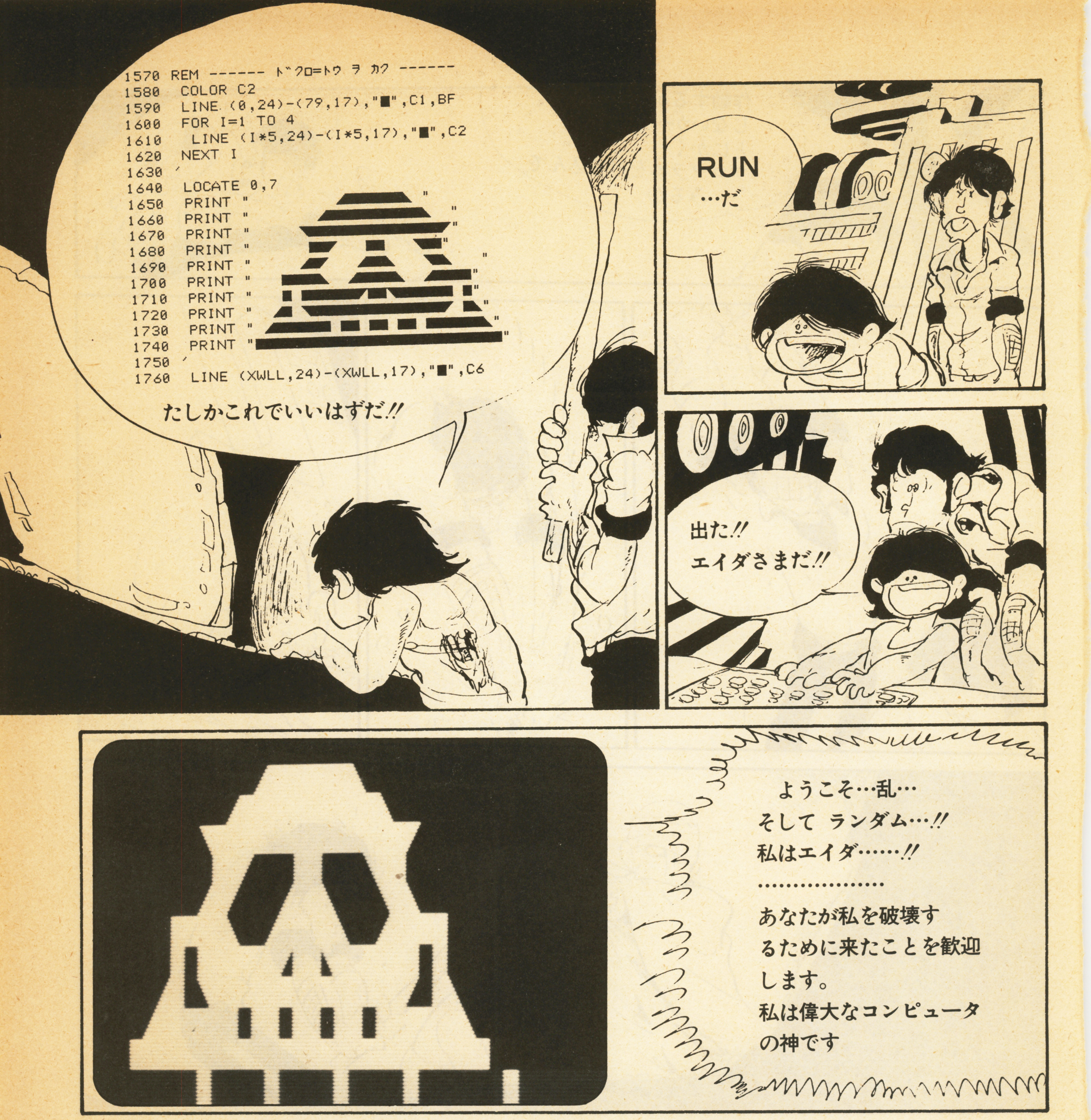
まず，キーが押されているかどうかを判断する。もし，どのキーも押されていなければ何も実行しない。押されていればそのキーによって「潜水艇の移動」・「レーザー砲の発射」・「ワープ」の３つのブロックのうちの１つを実行する。これらもブロックごとに作っておくといい。「レーザー砲の発射」ブロックでは，エネルギー・パイプが４本とも切断されたかどうか調べ，もし破壊されていれば「ドクロ島の崩壊」ブロックを呼び出し，ドクロ島を崩させるようにしよう。

2．ドクロ島側の処理

「防御シールドのすき間の移動」，「エネルギー・パイプの修復」ブロックが必要だ。対潜ヘリと爆雷については「対潜ヘリの移動」，「爆雷の投下」，「爆雷の落下」の３つのブロックを考えればいい。詳しいことは後で説明しよう。「爆雷の落下」ブロックで爆雷を落下させるわけだが，潜水艇にあたれば「潜水艇の爆発」ブロックを呼び出すことになる。

3．時間切れのチェック

ここは簡単だ。残り時間をチェックして時間切れでなければ１の潜水艇の処理に戻り，時間切れだったら「ゲーム・オーバー」ブロックに飛ぶようにする。



エイダを表示させる

PRINT文と グラフィックのキャラクタ

ついに，レムはプログラムを完成した。レムが叫ぶ。「RUNだ!」 音声合成*による機械的なエイダの声がマイコンのスピーカーから流れ出してくる。「ようこそ乱!…」さっそくリストを見ていくことにしよう。

ドクロ島と海の表示は1570〜1760行で処理している。1780行に RETURN 文があるのは，このプログラムの部分がサブルーチンになっているからだ。わかるね。

1590行の LINE 文が海の表示部分だ。四角を表示する場合には PSET 指定によるドット単位の LINE 文を使うことが多いのだが，ここでは後から GET @ 文を使ってキャラクタ単位でデータを読み込むためにキャラクタ単位の LINE 文を使っている。1600〜1620行の FOR〜NEXT 文と LINE 文でドクロ島のエネルギー・パイプを描いている。FOR 文が1回ループ*するたびにエネルギーパイプの X 座標が5キャラクタ分増加しているのがわかるかな。こうして4本のエネルギー・パイプが表示されるわけだ。

次の1640〜1740行でドクロ島の本体を描いている。タイトルの表示と同じようにグラフィック・キャラクタをうまく使っている。絵に自信があるのなら自由に書き換えてみるのも面白いね。ドクロ島は画面の左端に表示するので，タイトルの表示と違って TAB 関数が使われていない。

最後の1760行の LINE 文は防御シールドの表示用だ。

* 音声合成：人工的に人間の声を作ること。いま，盛んに研究されている。



移植のポイント

FM-8，PC-8001にあるキャラクタ単位のLINE文がPASOPIA，LEVEL Ⅲ，MZ-700にはない。そこでLOCATE文とPRINT文を使って表示している。このときFOR～NEXT文をうまく使えば少ないコマンドで処理できるわけだ。

また，PASOPIAのCOLOR文には注意してほしい。キャラクタ単位で色を指定するとき，Y座標が同じだとその行はすべて最後に指定された色になってしまう。つまり，横1列は必ず同じ色になってしまうわけだ。そこで，CHR$(248)～CHR$(255)を使って色とその範囲を指定して解決することになる。

```
1570 REM ------ ドクロ=トウ ヲ カク ------
1580  COLOR C2
1590  LINE (0,24)-(79,17),"■",C1,BF
1600  FOR I=1 TO 4
1610   LINE (I*5,24)-(I*5,17),"■",C2 *
1620  NEXT I
1630 '
1640  LOCATE 0,7
1650  PRINT "                    "
1660  PRINT "                    "
1670  PRINT "                    "
1680  PRINT "                    "
1690  PRINT "                    "
1700  PRINT "                    "
1710  PRINT "                    "
1720  PRINT "                    "
1730  PRINT "                    "
1740  PRINT "                    "
1750 '
1760  LINE (XWLL,24)-(XWLL,17),"■",C6
1770 '
1780  RETURN
```

＊ ループ：プログラムのなかでその流れが輪になっている部分。ここでは，FOR～NEXTループのこと。



エイダの出現 LOCATE, PRINT TAB関数

いよいよエイダのお出ましだ。ここでは，ゲームのタイトル表示の部分を説明することにしよう。

さあ，ラストアドベンチャー「ドクロ島破壊作戦」が始まる。この作戦に成功しなければ今までの苦労も水の泡となる。心をひきしめてチャレンジしよう。

RUNすると，画面には右のリストのようなタイトルが表示されるはずだ。さっそくプログラムのリストをのぞいてみよう。

タイトル表示は170～430行で処理している。COLOR 文，LOCATE 文，PRINT 文など，すでにおなじみのコマンド*で構成されているのがわかるね。「ドクロ＝トウ　ハカイサクセン」の表示は，グラフィックのキャラクタを組み合わせて，210～390行のPRINT 文で処理している。ここで，TAB(K)という関数を使っているが，これは文字を中央に表示させるためのテクニックのひとつだ。Kは桁の位置を示していて190行，270行，340行でKを指定することにより，3段のタイトルを中央にそろえている。試しに190行のK＝14をK＝0に変えてみよう。「ドクロ＝トウ」の表示だけが左端に移動する。つまり，桁位置をKという変数にしているため，タイトルの表示位置を自由に移動させることができるわけだ。

410～430行では，タイトルを囲む枠を表示している。LINE 文のオプション*であるBを忘れると，四角が表示されずに斜線になってしまう。注意しよう。

* コマンド：命令

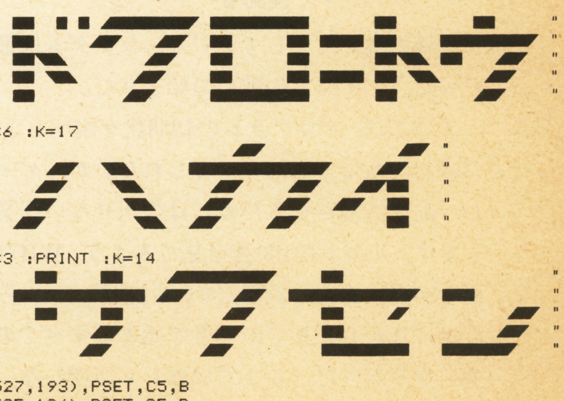
* オプション：自由に選択して使用できる部分。



移植のポイント

PRINT 文やLOCATE 文, TAB 関数は基本的にどのマイコンも同じだ。ただタイトルを囲む枠の表示処理をしているLINE 文が機種によって若干の違いがある。FM-8, LEVEL Ⅲ, PC-8001のLINE 文の機能(PSET, AND, OR など)がPASOPIA にはない。今回は問題ないけどね…。MZ-700にはLINE 文そのものがなく, スペースも足りないので, ここでは枠を描いていない。LINE 文でもうひとつ注意することがある。マイコンによってディスプレイのドット数が違う点だ。たとえば, PC-8001のドット数は160×100なのでX座標の値をFM-8(640×200) の1/4 にとらなくてはいけない。その点にさえ注意すれば問題ないだろう。

```
170 REM ----- ﾀｲﾄﾙ ﾋｮｳｼﾞ -----
180  COLOR C7 :LOCATE 32,2 :PRINT "Mission III :"
190  COLOR C1 :LOCATE 0,5 :K=14
200 '
210  PRINT TAB(K);"
220  PRINT TAB(K);"
230  PRINT TAB(K);"
240  PRINT TAB(K);"
250  PRINT TAB(K);"
260 '
270  PRINT :COLOR C6 :K=17
280  PRINT TAB(K);"
290  PRINT TAB(K);"
300  PRINT TAB(K);"
310  PRINT TAB(K);"
320  PRINT TAB(K);"
330 '
340  PRINT :COLOR C3 :PRINT :K=14
350  PRINT TAB(K);"
360  PRINT TAB(K);"
370  PRINT TAB(K);"
380  PRINT TAB(K);"
390  PRINT TAB(K);"
400 '
410  LINE (83,6)-(527,193),PSET,C5,B
420  LINE (75,3)-(535,196),PSET,C5,B
430  LINE (70,1)-(540,198),PSET,C5,B
```





初期設定 画面構成などを決める

相棒の乱をだましてまでも，ランダムは最初に出撃していった。エイダとの戦いの始まりだ。

ここでは，プログラムの初期化を説明しよう。ビギナー*にはわかりにくい部分かもしれないが，ゲームの場を設定し，乱を活躍させるためには欠かせない作業となる。

まず，リストの100行を見てみよう。WIDTH 80，25で画面を80桁×25行に設定している。次のコマンドDEFINT B-Zは，BからZまでの文字で始まるすべての変数を，整数型にする宣言だ。ここでは詳しく説明しないが，ゲームプログラムの変数は整数化することが多いと覚えておこう。最後のコマンドCONSOLE 0，25，0，0でスクロール行とファンクションキーの表示関係の指定をしている。

110〜120行は，プログラムで使う配列の宣言だ。配列Aは101個，配列BOMは1個，配列BMOLDは1個…と，データをそれぞれの個数だけ貯えることができるように宣言している。対潜ヘリや潜水艇の絵をデータとして貯えるための宣言と考えてもらえば間違いない。

150行のREAD文は4000行のDATA文と対応してカラーコードを変数で扱っている。こうしておけば，プログラム作成後に色を変えるのも楽だし，グリーン・ディスプレイではコードの近い色は区別がつきにくいので，その調節にも役立つ。

そして450〜460行で，対潜ヘリの高度や爆雷の数など，

＊ ビギナー＝beginner 初心者のこと



変数に初期値を代入している。

移植のポイント

初期設定では気をつけることがたくさんある。画面設定だが，MZ 700は40桁×25行の表示しかできないので，LOCATE文のX座標の値をすべてFM-8の1／2にすることが必要だ。

CONSOLE文は，マイコンごとにすべて違っている。この点は機種ごとのマニュアルを確認してほしい。とくにPC-8001では第4パラメータの意味（カラーモードあるいは単色モードの指定）がFM-8とまったく逆だから注意しよう。LEVEL Ⅲは初期設定のときにCONSOLE文だけでなくSCREEN文も使う。これはSCREEN関数とはまったく別のものだ。

変数名もここで決めているものが多いが，PC-8001は先頭2文字分しか判断しない。たとえば，変数PCAと変数PCBは同じ変数PCと判断してしまう。自分でプログラムを作るときには注意が必要だ。

```
100 WIDTH 80,25 :DEFINT B-Z :CONSOLE 0,25,0,0
110 DIM  A(100), BOM(0), BMOLD(0), FOOT(0), RHEL(6), LHEL(6)
120 DIM  SBM(7), SMOLD(7), WALL(2)
130 '
140 REM ----- ｶﾗｰ ﾃﾞｰﾀ ﾉ ﾖﾐｺﾐ -----
150  READ  C0, C1, C2, C3, C4, C5, C6, C7

450 REM ----- ﾍﾝｽｳ ﾉ ｼｮｷｶ ( 1ｶｲ ﾀﾞｹ ｽﾙ ) -----
460  YHEL=4 :NBMB=4 :XWLL=27 :YWLL=22 :FALSE=0 :TRUE=NOT FALSE

3990 REM ----- ｶﾗｰ ﾃﾞｰﾀ -----
4000  DATA  0, 1, 2, 3, 4, 5, 6, 7
```



ランダムの突撃 変数の設定とゲームのスタート

エイダに向かって突撃していくランダム。果してランダムはエイダを倒すことが出来るだろうか？

980～1040行で変数を設定している。対潜ヘリの位置と向きを決めているのが990～1000行。爆雷を投下可能にしているのが1030行。潜水艇の深度と位置を設定し，ワープが可能な状態にしているのが1010行だ。こうして，乱の潜水艇がスタンバイするといよいよ冒険がはじまる。

ゲームのスタート処理は1060～1120行だ。順番に見ていこう。1070行のLINE文で画面上方に白い線をひいて残り時間表示用のエリアと戦闘エリアを区別し，1080行で残り時間を表示している。残り時間の変数TMは1020のTM＝TM＋100で定義されているね。えっ，右辺のTMはどこにも定義されていないって。なにも定義していない変数は0であるということを忘れていないかな。つまり，初期の状態でTMは100になっているわけだ。このTMは3080～3170行のサブルーチンで1ずつ減って0になるとゲーム・オーバーだ。それまでにエネルギー・パイプを破壊し，エイダの息の根をとめなくてはいけない。

なお，TM＝0と最初に定義しないのは，ドクロ島が崩壊して次の面にすすむとき，残った時間に100を加えるためだ。

TMを定義しタイムを表示してから，1100行で「START」を表示し1200行で消しているが，その間の1110行でBEEP



文とFOR~NEXT文を使い出航のファンファーレ？を鳴らしている。

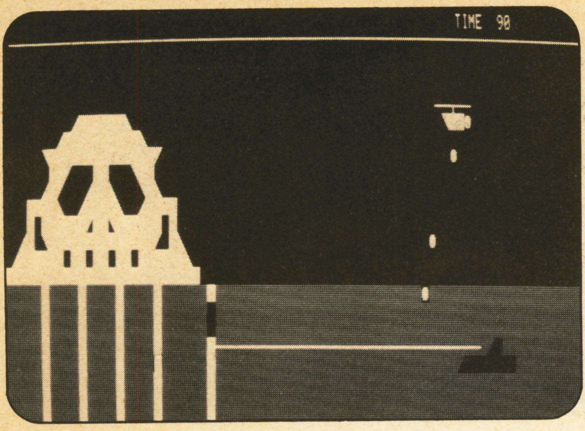
移植のポイント

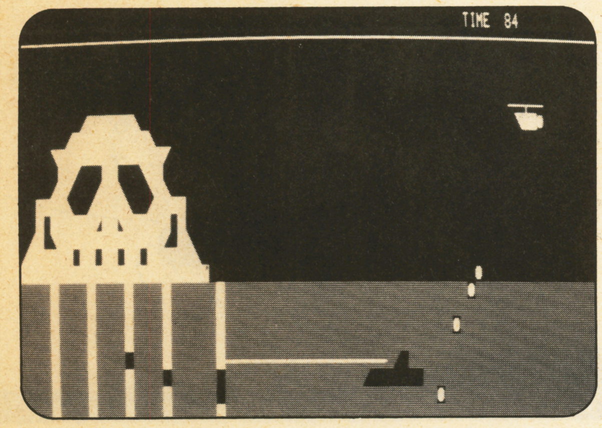
LEVEL Ⅲ にはBEEP 0, BEEP 1がない。そこでBEEPを何回か実行させてファンファーレの感じを出している。

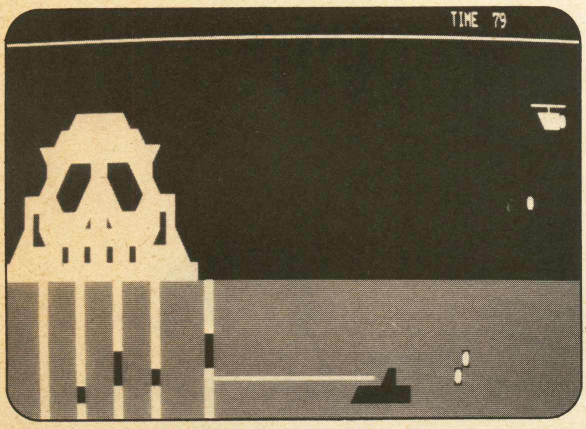
```
980 REM ------- ﾍﾝｽｳ ﾉ ｼｮｷｶ ( ｸﾘｶｴｼ ｽﾙ ) -------
990  XHEL=50 :DR=1
1000  YHEL=YHEL+2 :IF YHEL>15 THEN YHEL=6
1010  XSBM=60 :YSBM=21 :XOLDSBM=60 :YOLDSBM=21 :WARP=TRUE
1020  TM=TM+100 :RANDOMIZE(TIME/8-&H7FFF)
1030  PBMB=1 :FOR I=1 TO NBMB :BRDY(I)=TRUE :NEXT I
1040  COLOR C2,0 :CLS :GOSUB 1570 :GOSUB 1800
     :' ﾄﾞｸﾛ=ﾄｳ ｦ ｶｸ, ｾﾝｽｲｶﾝ ｦ ｶｸ
```

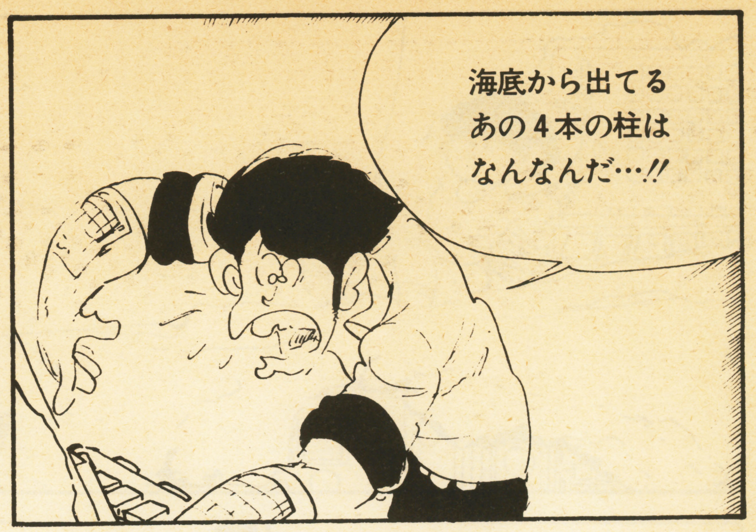
```
1060 REM ----- ｽﾀｰﾄ & ﾀｲﾑ -----
1070  LINE (0,20)-(639,20),PSET,C7
1080  COLOR C7 :LOCATE 60,1 :PRINT "TIME";TM
1090 '
1100  COLOR C6 :LOCATE 40,10 :PRINT "START"
1120  COLOR C0 :LOCATE 40,10 :PRINT "START" :COLOR C7
```

```
3080 REM ----- ｶｳﾝﾀｰ -----
3090  COLOR C7
3100  TM=TM-1
3110  LOCATE 65,1 :PRINT USING "###";TM;
3120  IF TM<>0 THEN RETURN
3130   BEEP
3140   COLOR C6 :LOCATE 35,4 :PRINT "ｼﾞｶﾝｷﾞﾚ !!!"
3150   FOR K=1 TO 20 :BEEP 1 :FOR J=1 TO 13 :BEEP 0 :NEXT J
      :NEXT K
3160   GOSUB 3940
3170   RETURN 1290
```











エイダの全貌　絵のデータの定義

ドクロ島の海域がディスプレイに表示され，エイダの全貌が明らかになってきたぞ！

戦闘の部分を見ていく前に，ゲームで使う絵のデータの定義部分を見ていくことにする。

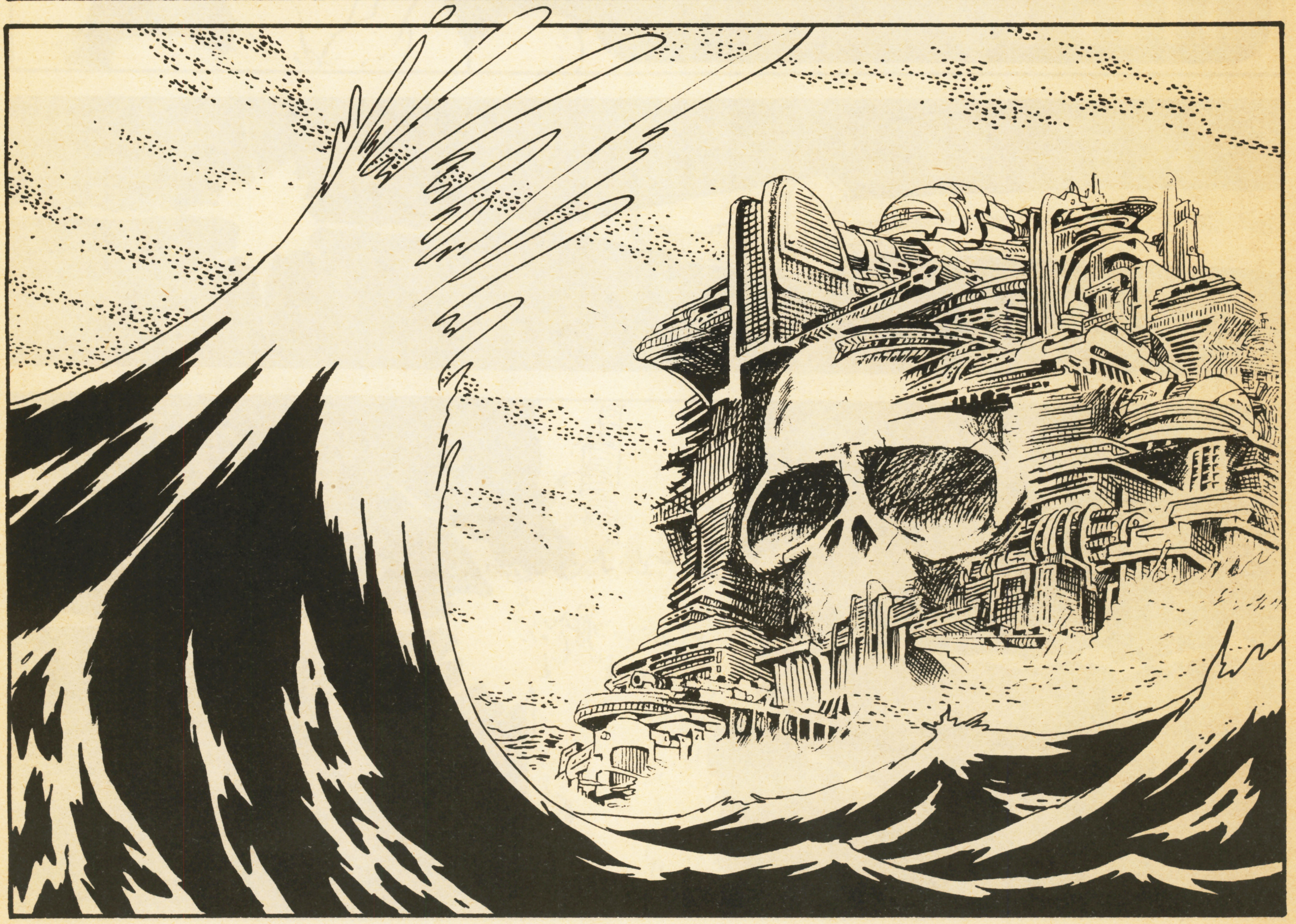
リストの490～920行を見てみよう。この部分で絵を表示し，そのデータを読み込んでいる。500～540行の対潜ヘリを例に説明しよう。まず対潜ヘリをLOCATE文とPRINT文で表示し，GET＠文でRHELという配列にキャラクタ単位で読み込む。こうすれば，PUT＠文を実行するだけで対潜ヘリを画面に表示できるようになる。もちろん，後のサブルーチンで移動範囲を指定するわけだが…。

ところで，520～530行を入力するときに，対潜ヘリの左右の空白を忘れないでほしい。空白部分で空(アキ)を表示することにより，その位置にあった対潜ヘリの絵を消しているわけだ。

乱がドクロ島を攻略するには，潜水艇に装備されたレーザー砲でエネルギー・パイプを破壊する必要がある。

そこで，防御シールドの唯一の弱点であるすき間の動きを解明しよう。プログラムは750～810行だ。XWLLがシールドのX座標，YWLLがシールドのすき間のY座標であり，その初期値は460行のXWLL＝27：YWLL＝22で決められている。770～800行までのPRINT文を見ればわかるように，780行と790行でシールドのすき間を表示し，GET＠



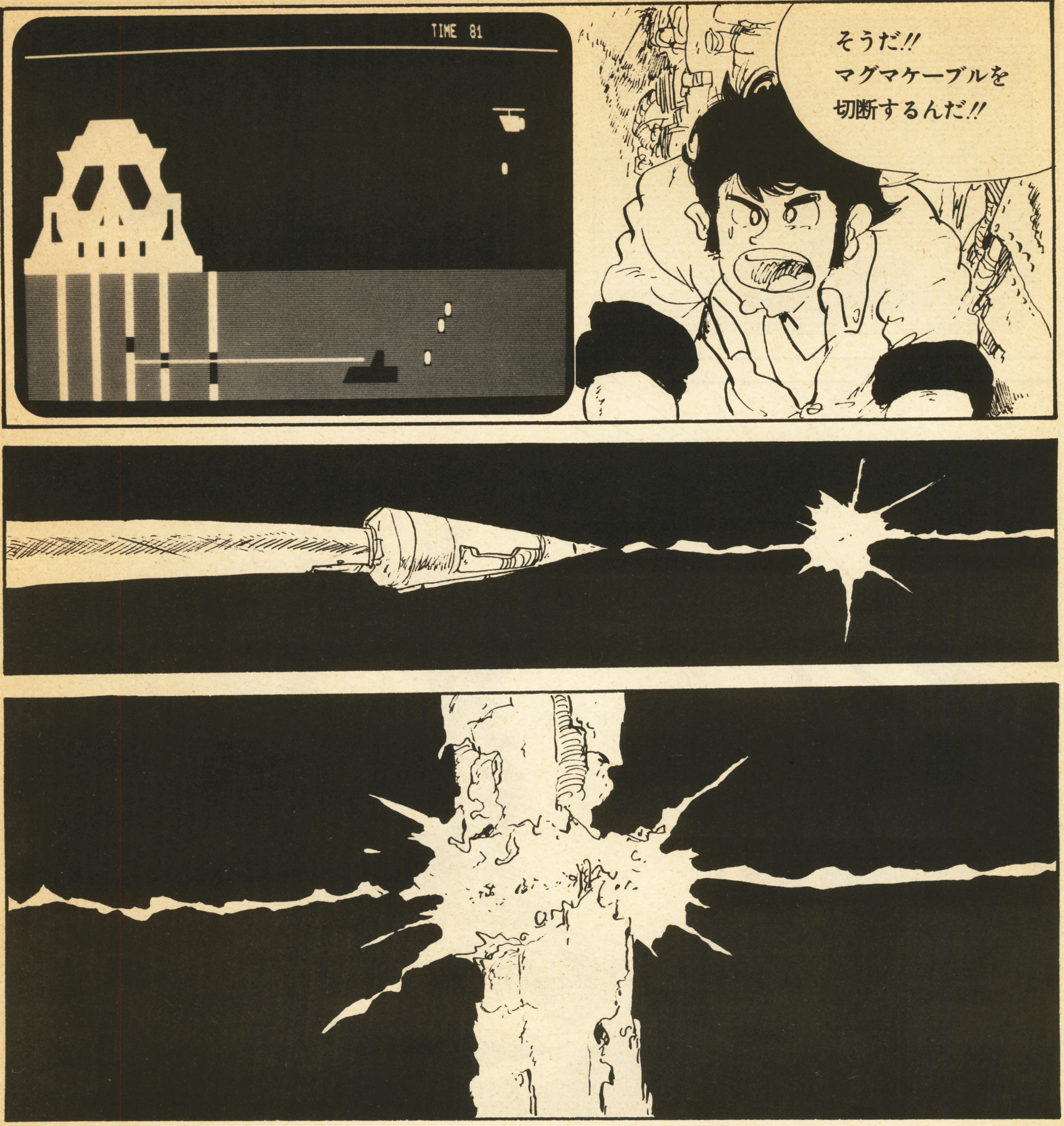


文で読み込むわけだ。読み込まれたデータは，後ろのサブルーチンのPUT @文で表示されることになる。

1140～1270行がメインルーチンだ。前半で入力されたキーを判断して潜水艇を移動させ，その後で「防御シールドのすき間の移動」,「エネルギー・パイプの修復」,「対潜ヘリの移動」,「爆雷の落下」,「時間の経過」の各サブルーチンを順に実行する。1270行のGOTO文により，この処理がくりかえされる。FM-8とPC-8001以外にはGET @文，PUT @文がないのでLOCATE文とPRINT文で代用。

```
500  REM  ﾍﾘｺﾌﾟﾀｰ (ﾐｷﾞ) ﾉ ﾃﾞｰﾀ
510   COLOR C4 :LOCATE 50,5 :PRINT "ﾍﾘｺﾌﾟﾀｰ"
520   LOCATE 50,7 :PRINT " ───┬─── "
530   LOCATE 50,8 :PRINT "  ◥■■▪● "
540   GET@ (50,7)-(56,8),RHEL
```

```
750  REM  ﾎﾞｳｷﾞｮ ﾓﾝ ﾉ ﾃﾞｰﾀ
760   COLOR C6 :LOCATE XWLL+2,YWLL :PRINT "ﾎﾞｳｷﾞｮﾓﾝ"
770   LOCATE XWLL,YWLL-1 :PRINT "■";
780   LOCATE XWLL,YWLL   :PRINT " ";
790   LOCATE XWLL,YWLL+1 :PRINT " ";
800   LOCATE XWLL,YWLL+2 :PRINT "■";
810   GET@ (XWLL,YWLL-1)-(XWLL,YWLL+2),WALL
1140 REM ------- ﾒｲﾝ ﾙｰﾁﾝ -------
1150  I$=INKEY$
1160   IF I$=" " THEN GOSUB 1870 :IF XBEEM=40 THEN
       GOSUB 2830 ELSE 1220
1170   IF I$="2" AND YSBM<23 THEN YSBM=YSBM+1 :GOSUB 1800
       :GOTO 1220
1180   IF I$="8" AND YSBM>17 THEN YSBM=YSBM-1 :GOSUB 1800
       :GOTO 1220
1190   IF I$="6" AND XSBM<70 THEN XSBM=XSBM+1 :GOSUB 1800
       :GOTO 1220
1200   IF I$="4" AND XSBM>40 THEN XSBM=XSBM-1 :GOSUB 1800
       :GOTO 1220
1210   IF I$="." THEN GOSUB 2060:' ﾜｰﾌﾟ
1220  GOSUB 2120:' ﾎﾞｳｷﾞｮ ﾓﾝ ﾉ ｶｲﾍｲ
1230  GOSUB 2190:' ﾄﾞｸﾛ=ﾄｳ ﾉ ｱｼ ｦ ﾉﾊﾞｽ
1240  GOSUB 2250:' ﾍﾘｺﾌﾟﾀｰ ｦ ｳｺﾞｶｽ
1250  GOSUB 2410:' ﾊﾞｸﾀﾞﾝ ｦ ｼﾀﾆ 1ﾂ ｽｽﾒﾙ
1260  GOSUB 3080:' ｼﾞｶﾝ ｦ 1 ﾍﾗｽ
1270 GOTO 1150
```



再生するエネルギー・パイプ GET@ PUT@

乱が叫ぶ。「そうだ！　エネルギー・パイプを破壊するんだ！」ランダムが発射したレーザーは，防御シールドのすき間をくぐりぬけて1本目のエネルギー・パイプを破壊した。乱が喜んだのもつかの間，切断したエネルギー・パイプもエイダの自己修復機能により再生されてしまう。

2120〜2170行と2190〜2230行で，防御シールドのすき間の移動と切断されたエネルギー・パイプを伸ばす処理をしている。

防御シールドのすき間の移動から説明しよう。2130行は比較的長い文になっているが，マルチステートメント*の前半の内容はWLLの定義だ。まずRND関数で発生した0から0.999999の間の数字を3倍し，0から2.999999の間の数字にしている。それをINT関数による整数化で0・1・2のいずれかにし，さらに1引いて −1・0・1のいずれかをWLLに代入している。つまり，乱数によって −1・0・1の整数を発生させているわけだ。そして後半のIF文で，防御シールドのY座標の変数YWLLにWLLを加えた値が領域をはみださないかチェックし，はみださない場合には2150行で実際にYWLLとWLLを加えてYWLLを定義しなおしている。はみだす場合には2130行をくりかえすわけだ。画面での防御シールドのすき間の移動は2160行のPUT@文が受け持っている。

次はエネルギー・パイプの修復を考えてみよう。リストは

*マルチステートメント=コロンで区切って1行に書かれた命令文

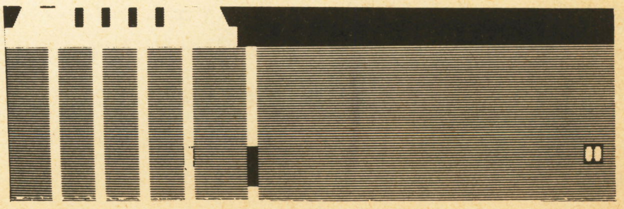


2190～2230行だ。まず2200行で3本のエネルギー・パイプのなかから1つを選びだす作業をしている。前ページのエネルギー・パイプの表示部分と前の防御シールドのすき間の移動処理の部分を参考にすれば理解できるね。そして2210行と2220行のGET @ 文, PUT @ 文で読み込む領域と書き出す領域を1キャラクタ分だけ下にずらして,エネルギー・パイプが下に伸びる表現をしているわけだ。2200行で発生する乱数によっては,破壊されていないエネルギー・パイプを下に伸ばすこともあるが,このとき外見上は何も変化しない。

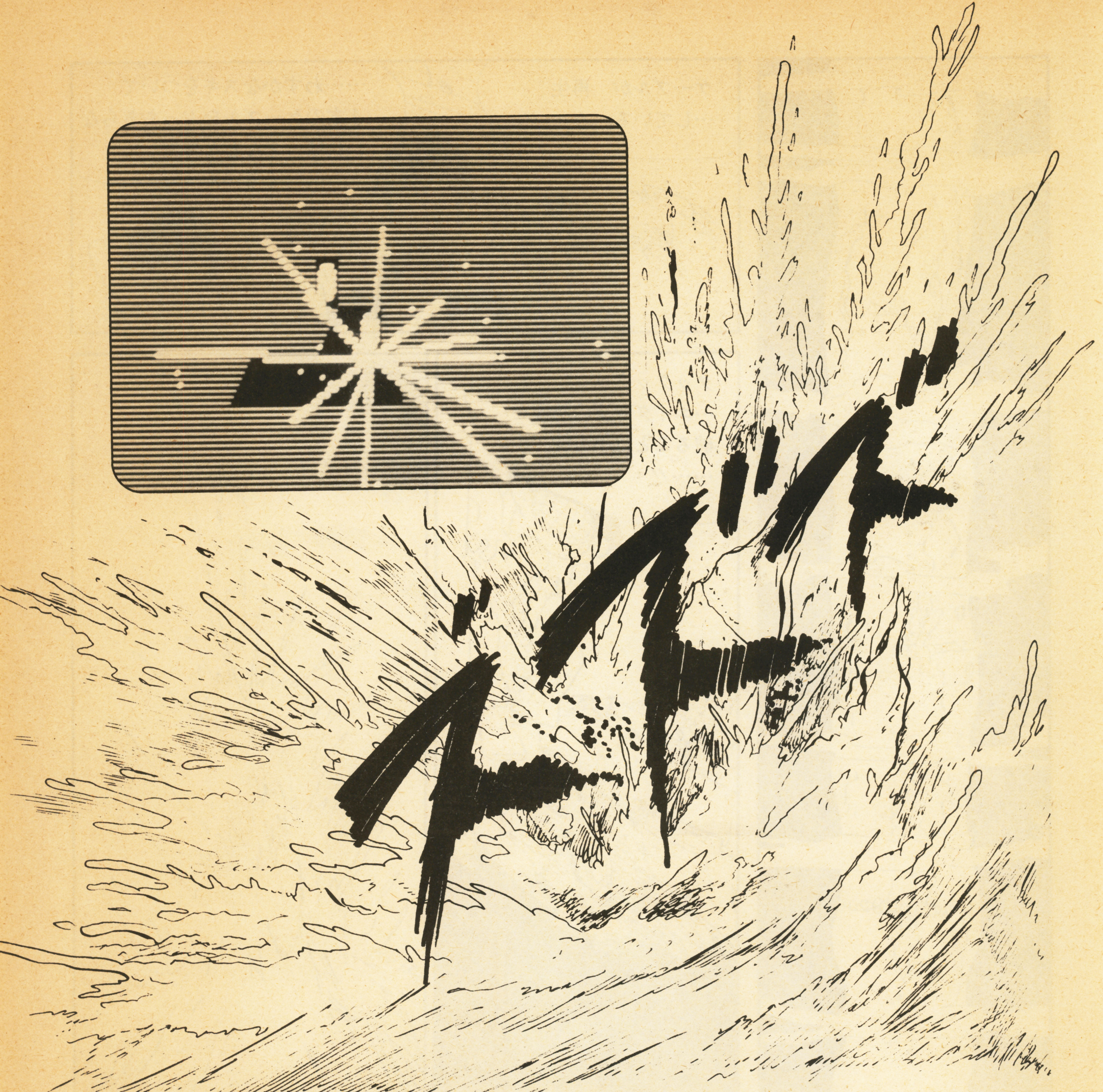
移植のポイント

ここでもGET @文, PUT @文の代わりにLOCATE文, PRINT文を使っている。

```
2120 REM ------- ホ゛ウキ゛ョ モン ノ カイヘイ -------
2130  WLL=INT(RND(1)*3)-1 :IF ( YWLL+WLL>22 )
      OR ( YWLL+WLL<18 ) THEN 2130
2140  '
2150  YWLL=YWLL+WLL
2160  PUT@ (XWLL,YWLL-1)-(XWLL,YWLL+2),WALL,C6
2170  RETURN
2190 REM ------- ト゛クロ=トウ ノ アシ -------
2200  FTILND=(INT(RND(1)*3+1))*5+5
2210  GET@ (FTILND,16)-(FTILND,23),A
2220  PUT@ (FTILND,17)-(FTILND,24),A,C2
2230  RETURN
```



▲防御門とエネルギー・パイプ



ランダムの最期 潜水艇の爆発

4本のエネルギー・パイプをすべて破壊しなくてはドクロ島を破壊できない。しかも，上空の対潜ヘリからは容赦なく爆雷が投下され，潜水艇の前後を気泡をたてながら落下していく。退避行動をとりながらレーザー砲の照準を合わせるため，ランダムは落ち着いてエネルギー・パイプを狙うことができない。しかし，ランダムの必死の攻撃にエネルギー・パイプは1本，2本と破壊されていく。「なんとしても，エネルギー・パイプが再生する前に4本とも破壊するんだ。」レーザーが海中に照射されるたびに，その光がランダムの顔を青白く照らしだす。「やった！ ついに3本目を破壊したぞ。あと1本だ。」ランダムの手が，再びレーザー砲の発射ボタンにかかったとき，潜水艇は大音響とともに四方に飛び散った。対潜ヘリの激しい爆雷攻撃に，さすがのランダムもあっけない最期だ。

爆雷にあたると潜水艇は爆発してゲーム・オーバーになる。この潜水艇の爆発シーンを考えてみることにしよう。

ディスプレイを見ると潜水艇に爆雷があたった場所を中心に放射状に線をランダム*にひき，赤と黄色の点を散りばめている。これは，プログラムではどうなっているのだろう？ リストを見てみよう。

2590～2630行はいつもの音出しだ。その次の2650～2670行で放射状の赤い線をひいている。X1，Y1には2550行で

＊ ランダム＝RANDOM でたらめに



計算した爆発の中心地が入っている。その値をもとに，2790～2810行のサブルーチンがランダムな点の位置を求めているわけだ。R＊SIN (A)，R＊COS*(A) で半径Rの円周上の1点を指定できるから，それに乱数関数RNDをかければ円内の1点を表すことができる。3がかけてあるのは水平方向のドットが垂直方向のドットより細かいからだ。FM-8はディスプレイのドット数が640×200だからね。

この部分がわかれば2690～2740行の点を打つブロックも理解できるね。最後の2760行は単なる時間待ちだ。

移植のポイント

ここでの問題は，ディスプレイのドット数の違いだけだ。

```
2570 REM ------- ﾊﾞｸﾊﾂ -------
2580   R=30
2590   FOR K=1 TO 50
2600    BEEP 1
2610    FOR J=1 TO 5 :NEXT J
2620    BEEP 0
2630   NEXT K
2640 '
2650   FOR A=0 TO 6.3 STEP .4
2660    GOSUB 2790 :LINE (X1,Y1)-(X,Y),PSET,C2
2670   NEXT A
2680 '
2690   FOR K=1 TO 2
2700    CC=C2 :IF K=2 THEN CC=C6 :R=37
2710    FOR A=0 TO 6.3 STEP .5
2720     GOSUB 2790 :PSET (X,Y,CC)
2730    NEXT A
2740   NEXT K
2750 '
2760  FOR I=1 TO 2000 :NEXT I
2770  RETURN 1290
2780 '
2790   REM ..... ｻﾞﾋｮｳ ﾖｳ ﾗﾝｽｳ .....
2800    X=R*RND(1)*COS(A)*3+X1 :Y=R*RND(1)*SIN(A)+Y1
2810    GOSUB 3190 :RETURN
```

＊ SIN COS＝三角関数。円の表示などによく使用する。



対潜ヘリの勝利

対潜ヘリの動き

はるか海上での爆発でランダムの最期を知った乱。その上空では勝ち誇ったように対潜ヘリが旋回している。

対潜ヘリの移動は2270～2320行だ。変数DRには対潜ヘリが右向きのときに1が，左向きのときに－1が入る。2270行で対潜ヘリのX座標を変え，2280～2290行では進行方向に合った絵をPUT@文で書き出している。2310～2320行では対潜ヘリを方向転換させて往復の動きを作っている。

2340～2380行は爆雷の投下だ。爆雷が使える状態の時は50%の確立で投下している。2370行で，投下した爆雷が落下中であることを配列BRDY( )に覚えさせているわけだ。2350～2360行で爆雷の投下位置を対潜ヘリの真下に設定している。

1度投下された爆雷は2410～2500行と2520～2550行で落下していく。2430行は爆雷の落下中かどうかの判断だ。配列BRDY(I)が－1ならばI番目の爆雷は投下されていないので何も処理しない。2440行は落下中の爆雷が海中か空中かの判断だ。爆雷のY座標が16より下ならば爆雷は海中に入っていく。

海中では，爆雷の落ちてゆく場所が青くぬられていれば(海の表示色)，さらに落下を続ける。そうでなければ潜水艇にあたったことになり，潜水艇の爆発ブロックへすすむ。当たったかどうかのチェックは2520～2540行の担当だ。



落下していった爆雷が海底まで到達した場合には，2470行で再び投下可能の状態にしている。配列BRDY（ ）の働きが理解できていれば，この行の意味はすぐわかるはずだ。

移植のポイント

レーザーが防御シールドやエネルギー・パイプにあたったかどうかの判断はSCREEN関数で行っているが，SCREEN関数のないPC-8001では，V-RAM*を直接PEEKするサブルーチンを追加して，そこでキャラクタの形を判断している。

```
2250 REM ----- ﾍﾘｺﾌﾟﾀｰ ----
2260  IF RND(1)>.85 THEN 2410
2270   XHEL=XHEL+DR
2280   IF DR=1 THEN PUT@ (XHEL,YHEL)-(XHEL+6,YHEL+1),RHEL,C4
       :GOTO 2310
2290    PUT@ (XHEL,YHEL)-(XHEL+6,YHEL+1),LHEL,C4
2310   IF XHEL>70 THEN DR=-1 :GOTO 2340
2320   IF XHEL<40 THEN DR=1
2330 '
2340  IF ( NOT BRDY(PBMB) ) OR ( RND(1)>.5 ) THEN 2390
2350   XB(PBMB)=XHEL+3
2360   YB(PBMB)=YHEL+2
2370   BRDY(PBMB)=FALSE
2380   PBMB=PBMB+1 :IF PBMB>NBMB THEN PBMB=1
2390  RETURN
2400 '
2410 REM ----- ﾊﾞｸﾀﾞﾝ ｺｳｶ -----
2420  FOR I=1 TO NBMB
2430   IF BRDY(I) GOTO 2490
2440    IF YB(I)>16 THEN 2520 ELSE COLOR C0
2450    PUT@ (XB(I),YB(I))-(XB(I),YB(I)),BMOLD
2460    YB(I)=YB(I)+1
2470    IF YB(I)=24 THEN BRDY(I)=TRUE :GOTO 2490
2480     PUT@ (XB(I),YB(I))-(XB(I),YB(I)),BOM,C2
2490  NEXT I
2500  RETURN
2510 '
2520 REM ----- ﾊﾞｸﾀﾞﾝ ｺｳｶ (ﾐｽﾞﾉﾅｶ) -----
2530  COLOR C1
2540  IF CHR$( SCREEN( XB(I),YB(I)+1 ) )="■" THEN 2450
2550  X1=XB(I)*8+4 :Y1=YB(I)*8+8
```

＊ V-RAM＝VIDEO-RAM（ビデオ ラム）の略。画面情報を貯えておくメモリ。



乱の出撃 潜水艇の処理

タブから買った潜水艇に乗って，乱が出撃していった。

潜水艇の動きは，1150行で入力されたキーに従って1160～1210行のIF 文で決定している。実際に各サブルーチンに飛ばしているのが1220～1260行のGOSUB 文だ。

1160行でレーザー砲サブルーチンを呼び出し，1870～2040行で実際に実行している。

1880～1890行では，キャラクタ単位の座標をドット単位の座標に変換するために8をかけている。次の1910～1950行でレーザーが防御シールドのすき間を通っているか判断し，通っていなければ潜水艇から防御シールドまでのレーザーを描いている。レーザーが防御シールドのすき間を通過するとプログラムは1970～2040行に移る。レーザーが当たるエネルギー・パイプを探し出し，そこまでレーザーを描くわけだ。

1170～1200行では潜水艇が領域からはみ出さないかチェックして，移動させている。GOSUB 1800は潜水艇の表示サブルーチンの呼び出しだ。

1210行ではワープのサブルーチンを呼び出している。2060～2110行のワープ処理を見ていこう。このサブルーチンが呼び出されると変数WARP が2080行で0になり，再びワープできなくなる。2090行がワープの秘密だ。ここで潜水艇の上に海を書くので，潜水艇はSCREEN 関数にひっかからなくなり，爆雷は潜水艇を素通りしてしまう。



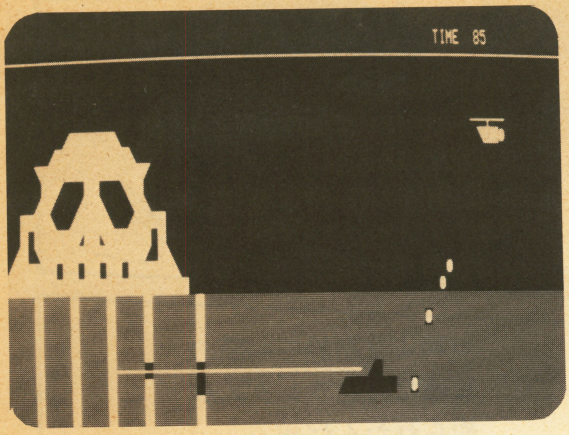
移植のポイント

キャラクタとドットの変換は機種ごとに違う。FM-8は8×8, PC-8001は2×4, MZ-700は2×2。PC-8001とMZ-700ではレーザーをキャラクタで描いていることにも注意しよう。

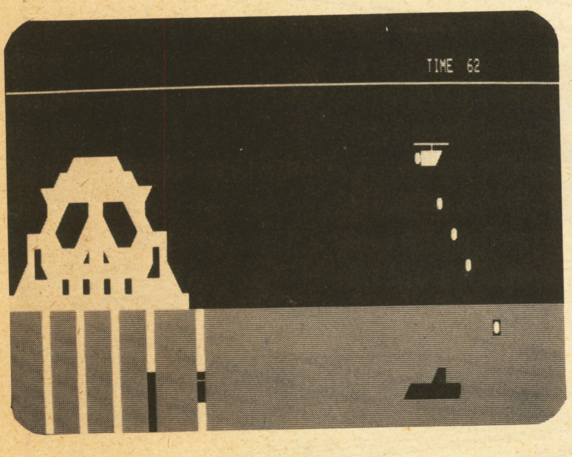
また, PC-8001では, 爆雷が潜水艇にあたったかどうかの判断をV-RAMを直接PEEKしている。

```
1800 REM ------- ｾﾝｽｲｶﾝ -------
1810  PUT@ (XOLDSBM,YOLDSBM)-(XOLDSBM+7,YOLDSBM+1),SMOLD,C1
1820 '
1830  PUT@ (XSBM,YSBM)-(XSBM+7,YSBM+1),SBM,C1
1840  XOLDSBM=XSBM :YOLDSBM=YSBM
1850  RETURN

1870 REM ------- ﾋﾞｰﾑ -------
1880  XBEEM=XWLL*8
1890  YBEEM=YSBM*8
1900 '
1910  IF CHR$( SCREEN( XWLL,YSBM ) )=" " THEN XBEEM=XBEEM-56
:GOTO 1970 ELSE XBE  EM=XBEEM+8
1920   LINE ((XSBM+3)*8,YBEEM+4)-(XBEEM,YBEEM+4),PSET,C6
1930   BEEP
1940   LINE ((XSBM+3)*8,YBEEM+4)-(XBEEM,YBEEM+4),PSET,C1
1950   RETURN
1960 '
1970   IF CHR$( SCREEN( XBEEM/8,YSBM ) )=" " THEN
       XBEEM=XBEEM-40 :GOTO 1970
1980    LINE ((XSBM+3)*8,YBEEM+4)-(XBEEM,YBEEM+4),PSET,C6
1990    BEEP
2000    LINE ((XSBM+3)*8,YBEEM+4)-(XBEEM-8,YBEEM+4),PSET,C1
2010    FOR I=20 TO XBEEM/8 STEP -5
2020     PUT@ (I,YBEEM/8)-(I,YBEEM/8),FOOT,C0
2030    NEXT I
2040    RETURN
2050 '
2060 REM ----- ﾜｰﾌﾟ -----
2070  IF NOT WARP THEN RETURN
2080   WARP=FALSE
2090   PUT@ (XSBM,YSBM)-(XSBM+7,YSBM+1),SMOLD,C1
2100   RETURN
2110
```









乱とエイダの死闘 崩壊シーンの作成

乱とエイダとの死闘は続く…。エネルギー・パイプをすべて破壊して，ドクロ島を破壊するんだ，乱!

この部分は2830～3060行のサブルーチンが受け持っている。2840～2910行でドクロ島の右側から3キャラクタずつの幅で崩壊の範囲を拡げながら徐々に下に崩す。つまり，最初は右端から3キャラクタ分の幅だが，次からは6キャラクタ分，9キャラクタ分，…と二重のFOR～NEXT文を使って3キャラクタ分ずつ崩壊を拡げているわけだ。この処理を最後まで続けると時間がかかるので，ある程度左端まで崩した後に2930～2990行でドクロ島全体を1キャラクタ分ずつ下に崩していく。

ドクロ島が完全に崩壊したあとで3190行の音出しサブルーチンを50回繰り返し，崩壊の残響をのこしている。

2840～2910行で大切なのはFOR～NEXT文の二重ループ*だ。内側のFOR～NEXT文が1回ループするたびに，2870行のGOSUB文により3190～3230行の音出しサブルーチンに飛ぶのでドクロ島は音を出しながらリアルに崩壊することになる。

次はゲーム・オーバーのサブルーチンを説明しよう。1300～1470行が「GAME OVER」の表示処理だが，この部分はタイトル表示の部分とまったく同じことをしている。詳しいことはタイトルの表示で説明した通りだ。

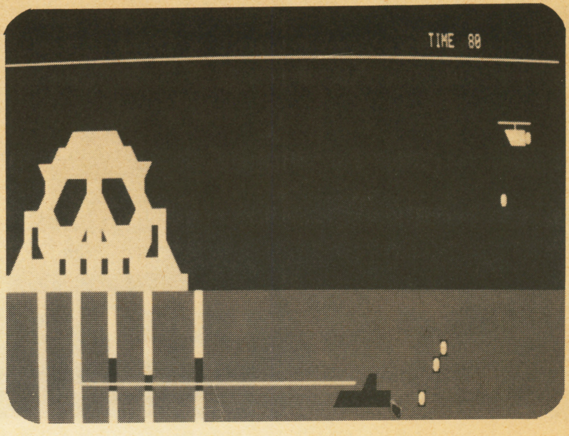
次の1490行は時間かせぎのためのFOR～NEXT文だね。

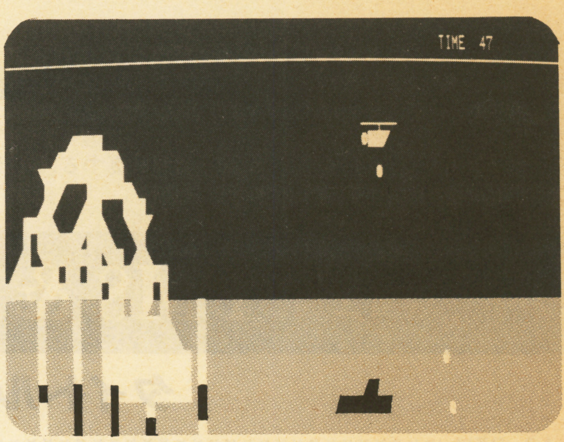
＊ 二重ループ＝二重になったループ。入れ子構造になっていなくてはいけない。

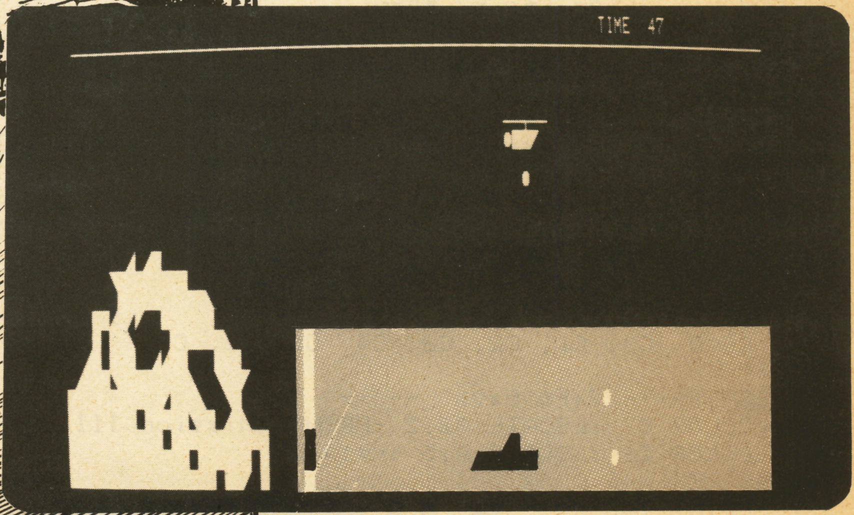
「乱」は読者であるキミだ…!!
……マグマケーブルを破壊して
エイダを海底に葬るか…!?
「乱」であるキミが海底のモクズとなるか…!?
……この結末をきめるのはキミだ…!!

RUNだ!! 乱!!









1520～1540行は再ゲームをするかどうかの確認だ。1530行で3250～3280行のInkeyルーチンに飛ぶ。このサブルーチンで，マイコンは君がなにかを入力するまで待ち続けるわけだ。君の入力が「Y」か「y」なら，1530行の後半と1540行で必要な初期化が行われ再びゲームが始まる。

もし，それ以外の入力があった場合は1550行のEND文により，ゲーム終了だ。

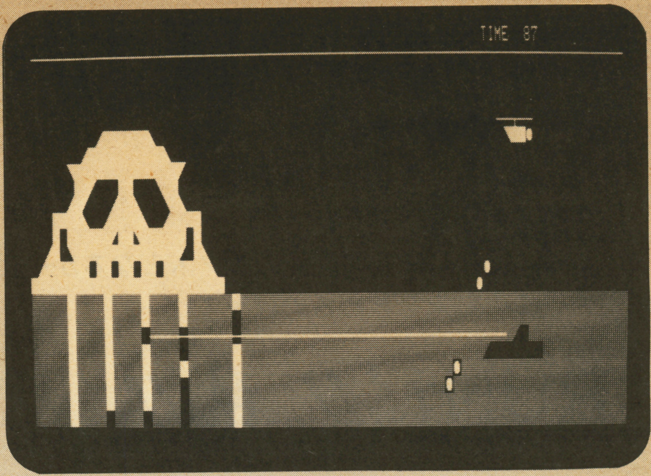
移植のポイント

ここでもGET＠文，PUT＠文が使えない機種は他のコマンドを代用している。DK$(9, 3)の配列にドクロ島のキャラクタを読み込んで，これをLOCATE文でずらして表示しているわけだ。

また，PC-8001ではINKEY$関数をそのまま使えない。

```
2830 REM ------- ホウカイ -------
2840  COLOR C2
2850  FOR I=23 TO 5 STEP -3
2860   FOR XILND=23 TO I STEP -3
2870    GOSUB 3190
2880    GET@ (XILND,6)-(XILND+2,21),A
2890    PUT@ (XILND,7)-(XILND+2,22),A
2900   NEXT XILND
2910  NEXT I
2920 '
2930  FOR YILND=8 TO 23
2940   GOSUB 3190
2950   GET@ (0,YILND)-(24,23),A
2960   GOSUB 3190
2970   PUT@ (0,YILND+1)-(24,24),A
2980   GOSUB 3190
2990  NEXT YILND
3000 '
3010  FOR I=1 TO 50
3020   GOSUB 3190
3030  NEXT I
3040 '
3050  FOR I=1 TO 100 :NEXT I
3060  RETURN 980
```





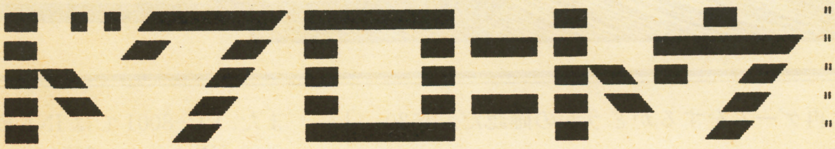
レーザー・サブマリン

●遊び方　潜水艇を操縦してドクロ島を破壊する。詳しいことは126ページの「遊び方の説明」を読んでほしい。

(富士通) FM8 & FM7全リスト

タイトルの表示と初期設定

```
10 REM ************************************************
20 REM *                                              *
30 REM *     ﾄﾞｸﾛ=ﾄｳ ﾊｶｲ ｻｸｾﾝ                           *
40 REM *                                              *
50 REM *        Programmed  by  ETX                   *
60 REM *                                              *
70 REM *         Special thanks to  Nor & RAN         *
80 REM ************************************************
90 '
100 WIDTH 80,25 :DEFINT B-Z :CONSOLE 0,25,0,0
110 DIM  A(100), BOM(0), BMOLD(0), FOOT(0), RHEL(6), LHEL(6)
120 DIM  SBM(7), SMOLD(7), WALL(2)
130 '
140 REM ----- ｶﾗｰ ﾃﾞｰﾀ ﾉ ﾖﾐｺﾐ -----
150  READ  C0, C1, C2, C3, C4, C5, C6, C7
160 '
170 REM ----- ﾀｲﾄﾙ ﾋｮｳｼﾞ -----
180  COLOR C7 :LOCATE 32,2 :PRINT "Mission III :"
190  COLOR C1 :LOCATE 0,5 :K=14
200 '
210  PRINT TAB(K);"
220  PRINT TAB(K);"
230  PRINT TAB(K);"
240  PRINT TAB(K);"
250  PRINT TAB(K);"
260 '
270  PRINT :COLOR C6 :K=17
280  PRINT TAB(K);"
290  PRINT TAB(K);"
300  PRINT TAB(K);"
310  PRINT TAB(K);"
320  PRINT TAB(K);"
330 '
340  PRINT :COLOR C3 :PRINT :K=14
350  PRINT TAB(K);"
360  PRINT TAB(K);"
370  PRINT TAB(K);"
380  PRINT TAB(K);"
390  PRINT TAB(K);"
400 '
```







画面の設定と変数の型指定 (100)

必要な配列の宣言 (110–120)

色コードを変数に読み込む (150)

変　数　表　❶

〔変　数〕	
A	爆発用一時変数
ANS$	解答判定用文字変数
BRDY( )	爆雷状態変数
C0〜C7	カラーコード
CC	爆発用色変数
DR	ヘリコプターの方向
TRUE, FALSE	状態判定用変数
I, J, K	FOR〜NEXT 等の一時変数
I$	入力キー
NBMB	爆雷の数
PBMB	落下爆雷を指し示す変数
R	円の半径
TM	タイム
WLL	防御門移動方向
WARP	ワープを使ったかの判定用変数

変　数　表　❷

〔GET@, PUT@ 用配列〕	
A( )	ドクロ島崩壊用
BOM( )	爆雷
BMOLD( )	爆雷消去用
FOOT( )	パイプ
LHEL( )	左向きヘリコプター
RHEL( )	右向きヘリコプター
SBM( )	潜水艇
SMOLD( )	潜水艇消去用
WALL( )	防御門
〔座標用変数〕	
FTILND	のばすパイプのX座標
X, Y	爆発用一時変数
XI, YI	爆発の中心
XB( ), YB( )	爆雷の位置
XBEEM, YBEEM	ビームの到達位置
XHEL, YHEL	ヘリコプターの位置
XILND, YILND	崩壊時のGET@, PUT@の範囲
XOLDSBM, YOLDSBM	移動前の潜水艇の位置
XSBM, YSBM	現在の潜水艇の位置
XWLL, YWLL	防御門の位置

```
410  LINE (83,6)-(527,193),PSET,C5,B  ┐
420  LINE (75,3)-(535,196),PSET,C5,B  ├──── タイトルを四角で囲む
430  LINE (70,1)-(540,198),PSET,C5,B  ┘
440 '
450 REM ----- ヘンスウ ノ ショキカ ( 1カイ ダケ スル ) -----        値を変えない変数の値の設定
460  YHEL=4 :NBMB=4 :XWLL=27 :YWLL=22 :FALSE=0 :TRUE=NOT FALSE ──┘
470  FOR I=1 TO 1500 :NEXT I :CLS :GOSUB 1570 :LOCATE 10,5 :PRINT "ドクロ=トウ"
480 '
```

データの読み込み

```
490 REM ------ PUT@ ヨウ データ ノ ヨミコミ -----
500  REM  ヘリコプター (ミギ) ノ データ
510   COLOR C4 :LOCATE 50,5 :PRINT "ヘリコプター"
520   LOCATE 50,7 :PRINT " ──┬── "
530   LOCATE 50,8 :PRINT "  ◥■■● "
540   GET@ (50,7)-(56,8),RHEL──────────── 右向き対潜ヘリデータの読み込み
550 '
560  REM  バクダン (モト ノ バショ) ノ データ
570   COLOR C0
580   LOCATE 53,9 :PRINT "■"
590   GET@ (53,9)-(53,9),BMOLD──────────── 爆雷を消すためのデータの読み込み
600 '
610  REM  バクダン ノ データ
620   COLOR C2
630   LOCATE 53,10 :PRINT "●";"   バクダン"
640   GET@ (53,10)-(53,10),BOM──────────── 爆雷のデータの読み込み
650 '
660  REM  ウミ (センスイカン ガ モト イタトコロ) ノ データ
670   GET@ (60,21)-(67,22),SMOLD──────────── 潜水艇を消すためのデータの読み込み
680 '
690  REM  センスイカン ノ データ
700   LOCATE 60,19 :PRINT "センスイカン" :COLOR C1
710   LOCATE 60,21 :PRINT "■■■◤ ■■";
720   LOCATE 60,22 :PRINT "◤      ";
730   GET@ (60,21)-(67,22),SBM──────────── 潜水艇のデータの読み込み
740 '
750  REM  ボウギョ モン ノ データ
760   COLOR C6 :LOCATE XWLL+2,YWLL :PRINT "ボウギョモン"
770   LOCATE XWLL,YWLL-1 :PRINT "■";
780   LOCATE XWLL,YWLL   :PRINT " ";
790   LOCATE XWLL,YWLL+1 :PRINT " ";
800   LOCATE XWLL,YWLL+2 :PRINT "■";
810   GET@ (XWLL,YWLL-1)-(XWLL,YWLL+2),WALL──────────── 防御門のデータの読み込み
820 '
830  REM  ドクロ=トウ ノ アシガ, ハカイ サレタ トコロ ノ データ
840   COLOR C2 :LOCATE 22,19 :PRINT "アシ"
850   LOCATE 20,20 :PRINT " "
860   GET@ (20,20)-(20,20),FOOT──────────── パイプのすきまのデータの読み込み
870 '
880  REM  ヘリコプター (ヒダリ) ノ データ
890   COLOR C4
900   LOCATE 50,7 :PRINT " ──┬── "
910   LOCATE 50,8 :PRINT " ●■■◤  "
920   GET@ (50,7)-(56,8),LHEL──────────── 左向き対潜ヘリデータの読み込み
930 '
```

メインルーチン

```
940 REM ------ ｾﾂﾒｲ ﾊ ? ------
950  GOSUB 3940 :' ｼﾞｶﾝﾏﾁ ﾍ
960  COLOR C7 :CLS :GOSUB 3300
970 '
980 REM ------- ﾍﾝｽｳ ﾉ ｼｮｷｶ ( ｸﾘｶｴｼ ｽﾙ ) -------
990  XHEL=50 :DR=1
1000  YHEL=YHEL+2 :IF YHEL>15 THEN YHEL=6
1010  XSBM=60 :YSBM=21 :XOLDSBM=60 :YOLDSBM=21 :WARP=TRUE
1020  TM=TM+100 :RANDOMIZE(TIME/8-&H7FFF)
1030  PBMB=1 :FOR I=1 TO NBMB :BRDY(I)=TRUE :NEXT I
1040  COLOR C2,0 :CLS :GOSUB 1570 :GOSUB 1800 :' ﾄﾞｸﾛ=ﾄｳ ｦ ｶｸ, ｾﾝｽｲｶﾝ ｦ ｶｸ
1050 '
1060 REM ----- ｽﾀｰﾄ & ﾀｲﾑ -----
1070  LINE (0,20)-(639,20),PSET,C7
1080  COLOR C7 :LOCATE 60,1 :PRINT "TIME";TM
1090 '
1100  COLOR C6 :LOCATE 40,10 :PRINT "START"
1110  FOR I=1 TO 30 :BEEP1 :FOR J=1 TO 10 :NEXT J :BEEP0 :NEXT I
1120  COLOR C0 :LOCATE 40,10 :PRINT "START" :COLOR C7
1130 '
1140 REM ------- ﾒｲﾝ ﾙｰﾁﾝ -------
1150  I$=INKEY$
1160   IF I$=" " THEN GOSUB 1870 :IF XBEEM=40 THEN GOSUB 2830 ELSE 1220
1170   IF I$="2" AND YSBM<23 THEN YSBM=YSBM+1 :GOSUB 1800 :GOTO 1220
1180   IF I$="8" AND YSBM>17 THEN YSBM=YSBM-1 :GOSUB 1800 :GOTO 1220
1190   IF I$="6" AND XSBM<70 THEN XSBM=XSBM+1 :GOSUB 1800 :GOTO 1220
1200   IF I$="4" AND XSBM>40 THEN XSBM=XSBM-1 :GOSUB 1800 :GOTO 1220
1210   IF I$="." THEN GOSUB 2060:' ﾜｰﾌﾟ
1220  GOSUB 2120:' ﾎﾞｳｷﾞｮ ﾓﾝ ﾉ ｶｲﾍｲ
1230  GOSUB 2190:' ﾄﾞｸﾛ=ﾄｳ ﾉ ｱｼ ｦ ﾉﾊﾞｽ
1240  GOSUB 2250:' ﾍﾘｺﾌﾟﾀｰ ｦ ｳｺﾞｶｽ
1250  GOSUB 2410:' ﾊﾞｸﾀﾞﾝ ｦ ｼﾀﾆ 1ﾂ ｽｽﾒﾙ
1260  GOSUB 3080:' ｼﾞｶﾝ ｦ 1 ﾍﾗｽ
1270 GOTO 1150
1280 '
```

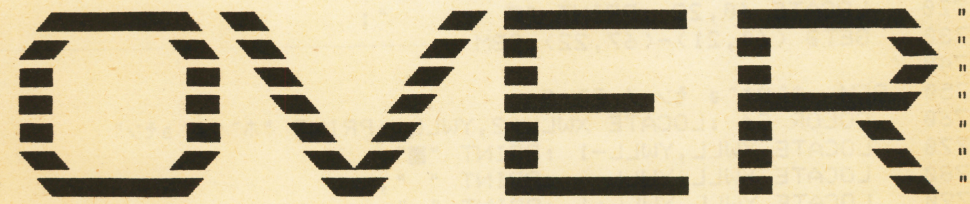
海面ぎりぎりにまでなったら再び上にもどす (1000)
潜水艇の初期位置決定；ワープを使えるようにする (1010)
残り時間決定 (1020)
爆雷を全部投下可能にする (1030)
残り時間表示 (1080)
「START」書き出し (1100)
音出し (1110)
「START」消す (1120)
ドクロ画崩壊サブルーチンへ (1160)
潜水艇表示サブルーチンへ (1200)

ゲームオーバー

```
1290 REM ----- End ﾉ ﾀｲﾄﾙ ﾋｮｳｼﾞ -----
1300  COLOR C6,C2 :CLS :K=12 :LOCATE 0,4
1310 '
1320  PRINT TAB(K);"
1330  PRINT TAB(K);"
1340  PRINT TAB(K);"
1350  PRINT TAB(K);"
1360  PRINT TAB(K);"
1370  PRINT TAB(K);"
1380  PRINT TAB(K);"
```



```
1390 '
1400  LOCATE 0,13
1410  PRINT TAB(K);"
1420  PRINT TAB(K);"
1430  PRINT TAB(K);"
1440  PRINT TAB(K);"
1450  PRINT TAB(K);"
1460  PRINT TAB(K);"
1470  PRINT TAB(K);"
```



```
1480 '
1490  FOR I=0 TO 1500 :NEXT I
1500  COLOR C7,0 :CLS
```

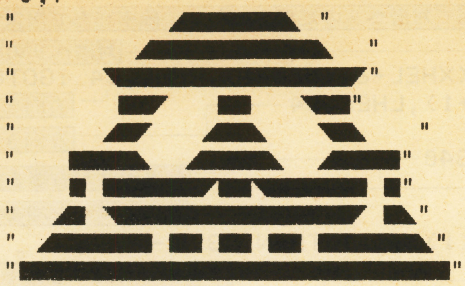
REPLAY

```
1510  REM ..... try again ? .....
1520   BEEP :PRINT :PRINT :PRINT "     ｻｲﾄﾞ ﾄﾗｲ ｽﾙｶ (Y/N) ?";
1530   GOSUB 3250 :ANS$=I$ :PRINT ANS$
1540   IF ( ANS$="Y" ) OR ( ANS$="y" ) THEN YHEL=4: TM=0: GOTO 980
1550   END
1560  '
```

キー入力を読んで画面に書き出す (1530)
もういちどあそぶのなら簡単なイニシャライズをしてから980行へ (1540)

ドクロ島と海の表示サブルーチン

```
1570 REM ------ ﾄﾞｸﾛ=ﾄｳ ｦ ｶｸ ------
1580  COLOR C2
1590  LINE (0,24)-(79,17),"■",C1,BF        ―― 海を描く
1600  FOR I=1 TO 4                         ┐
1610   LINE (I*5,24)-(I*5,17),"■",C2       ┘―― エネルギーパイプを描く
1620  NEXT I
1630 '
1640  LOCATE 0,7
1650  PRINT "
1660  PRINT "
1670  PRINT "
1680  PRINT "
1690  PRINT "
1700  PRINT "
1710  PRINT "
1720  PRINT "
1730  PRINT "
1740  PRINT "
```



```
1750 '
1760  LINE (XWLL,24)-(XWLL,17),"■",C6      ―― 防御シールドを描く
1770 '
1780  RETURN
1790 '
```

潜水艇の動きとレーザー砲サブルーチン

```
1800 REM ------- ｾﾝｽｲｶﾝ -------
1810  PUT@ (XOLDSBM,YOLDSBM)-(XOLDSBM+7,YOLDSBM+1),SMOLD,C1
1820 '                              └―― 潜水艇の元いた場所を青くぬりつぶす（海にもどす）
1830  PUT@ (XSBM,YSBM)-(XSBM+7,YSBM+1),SBM,C1 ―― 新しい潜水艇の位置に潜水艇の絵を描き出す
1840  XOLDSBM=XSBM :YOLDSBM=YSBM ―― 今の潜水艇の位置を覚えておく（次に消すときにつかう）
1850  RETURN
1860 '
1870 REM ------- ﾋﾞｰﾑ -------
1880  XBEEM=XWLL*8  ┐
1890  YBEEM=YSBM*8  ┘―― キャラクタ単位の座標をドット単位の座標に変換する
1900 '
1910  IF CHR$( SCREEN( XWLL,YSBM ) )=" " THEN XBEEM=XBEEM-56 :GOTO 1970 ELSE XBE
EM=XBEEM+8   └―― レーザー砲が防御シールドのすき間を通れば１番右のエネルギーパイプまでレーザー砲を進める。
                  そうでなければ防御シールドまで

1920   LINE ((XSBM+3)*8,YBEEM+4)-(XBEEM,YBEEM+4),PSET,C6 ―― レーザー砲の発射
1930   BEEP
1940   LINE ((XSBM+3)*8,YBEEM+4)-(XBEEM,YBEEM+4),PSET,C1 ―― レーザー砲の消去
1950   RETURN
1960 '                         ┌―― エネルギーパイプの穴を通れば次のエネルギーパイプまで進める
1970   IF CHR$( SCREEN( XBEEM/8,YSBM ) )=" " THEN XBEEM=XBEEM-40 :GOTO 1970
1980    LINE ((XSBM+3)*8,YBEEM+4)-(XBEEM,YBEEM+4),PSET,C6 ―― レーザー砲の発射
1990    BEEP
2000    LINE ((XSBM+3)*8,YBEEM+4)-(XBEEM-8,YBEEM+4),PSET,C1 ―― レーザー砲の消去
2010    FOR I=20 TO XBEEM/8 STEP -5          ┐
2020     PUT@ (I,YBEEM/8)-(I,YBEEM/8),FOOT,C0 ┘
2030    NEXT I ―― エネルギーパイプの穴を通ったところのレーザー砲の消去(こうしないと青い線が残る
2040    RETURN
2050 '
2060 REM ----- ﾜｰﾌﾟ -----
2070  IF NOT WARP THEN RETURN ―― 過去にワープを使っていれば何も処理しないですぐサブルーチンを出る
2080   WARP=FALSE ―― ワープを使ったことを覚えておく
2090   PUT@ (XSBM,YSBM)-(XSBM+7,YSBM+1),SMOLD,C1 ―┐
2100   RETURN                                     └―― 潜水艇の上に海を描き出す
2110 '
```

防御シールドとエネルギーパイプの動きサブルーチン

```
2120 REM ------- ﾎﾞｳｷﾞｮ ﾓﾝ ﾉ ｶｲﾍｲ -------
2130  WLL=INT(RND(1)*3)-1 :IF ( YWLL+WLL>22 ) OR ( YWLL+WLL<18 ) THEN 2130
2140  '                                       └―― もし領域からはみ出るならもう１ど決定し直す
2150  YWLL=YWLL+WLL ―― 防御シールドのＹ座標を決める
2160  PUT@ (XWLL,YWLL-1)-(XWLL,YWLL+2),WALL,C6 ―― 防御シールドの描き出し
2170  RETURN
2180 '
```

```
2190 REM ------- ﾄﾞｸﾛ=ﾄｳ ﾉ ｱｼ -------
2200  FTILND=(INT(RND(1)*3+1))*5+5
2210  GET@ (FTILND,16)-(FTILND,23),A
2220  PUT@ (FTILND,17)-(FTILND,24),A,C2
2230  RETURN
2240 '
```

- 2200: 修復するエネルギーパイプのえらび出し
- 2210: エネルギーパイプの絵のデータ読み込み
- 2220: パイプを1段さげて描き出す

対潜ヘリの動きと爆雷サブルーチン

```
2250 REM ----- ﾍﾘｺﾌﾟﾀｰ ----
2260  IF RND(1)>.85 THEN 2410
2270   XHEL=XHEL+DR
2280   IF DR=1 THEN PUT@ (XHEL,YHEL)-(XHEL+6,YHEL+1),RHEL,C4 :GOTO 2310
2290    PUT@ (XHEL,YHEL)-(XHEL+6,YHEL+1),LHEL,C4
2300 '
2310    IF XHEL>70 THEN DR=-1 :GOTO 2340
2320    IF XHEL<40 THEN DR=1
2330 '
2340   IF ( NOT BRDY(PBMB) ) OR ( RND(1)>.5 ) THEN 2390
2350    XB(PBMB)=XHEL+3
2360    YB(PBMB)=YHEL+2
2370    BRDY(PBMB)=FALSE
2380    PBMB=PBMB+1 :IF PBMB>NBMB THEN PBMB=1
2390   RETURN
2400 '
2410 REM ----- ﾊﾞｸﾀﾞﾝ ｺｳｶ -----
2420  FOR I=1 TO NBMB
2430   IF BRDY(I) GOTO 2490
2440    IF YB(I)>16 THEN 2520 ELSE COLOR C0
2450    PUT@ (XB(I),YB(I))-(XB(I),YB(I)),BMOLD
2460    YB(I)=YB(I)+1
2470    IF YB(I)=24 THEN BRDY(I)=TRUE :GOTO 2490
2480     PUT@ (XB(I),YB(I))-(XB(I),YB(I)),BOM,C2
2490  NEXT I
2500  RETURN
2510 '
2520 REM ----- ﾊﾞｸﾀﾞﾝ ｺｳｶ (ﾐｽﾞﾉﾅｶ) -----
2530  COLOR C1
2540  IF CHR$( SCREEN( XB(I),YB(I)+1 ) )="■" THEN 2450
2550  X1=XB(I)*8+4 :Y1=YB(I)*8+8
2560 '
```

- 2260: 確率85%で対潜ヘリの処理を行ない，確率15%で爆雷投下の処理を行なう
- 2270: 対潜ヘリの座標を1だけ動かす
- 2280: 右向き対潜ヘリの描き出し
- 2290: 左向き対潜ヘリの描き出し
- 2310: 対潜ヘリが右端までいくと左に向ける
- 2320: 対潜ヘリが左端までいくと右に向ける
- 2340: 爆雷投下可能なら確率50%で投下を行なう
- 2350–2360: 爆雷の投下位置の決定
- 2370: 落下中であることを覚えておく
- 2380: 落下の対象となる爆雷を1つ進める
- 2430: 1番目の爆雷が落下中でなければ2490行へ
- 2440: 爆雷がまだ海に届いていなければ2520行へ
- 2460: 今の爆雷のある場所を空中なら黒，海中なら青でぬる
- 2470: 爆雷が海底まで届いていれば再び投下可能にする
- 2480: 新しい場所に爆雷の絵を描き出す
- 2550: 爆発の中心位置の決定

潜水艇の爆発サブルーチン

```
2570 REM ------- ﾊﾞｸﾊﾂ -------
2580   R=30
2590   FOR K=1 TO 50
2600    BEEP 1
2610    FOR J=1 TO 5 :NEXT J
2620    BEEP 0
2630   NEXT K
2640 '
2650   FOR A=0 TO 6.3 STEP .4
2660    GOSUB 2790 :LINE (X1,Y1)-(X,Y),PSET,C2
2670   NEXT A
2680 '
2690   FOR K=1 TO 2
2700    CC=C2 :IF K=2 THEN CC=C6 :R=37
2710    FOR A=0 TO 6.3 STEP .5
2720     GOSUB 2790 :PSET (X,Y,CC)
2730    NEXT A
2740   NEXT K
2750 '
2760  FOR I=1 TO 2000 :NEXT I
2770  RETURN 1290
2780 '
2790   REM ..... ｻﾞﾋｮｳ ﾖｳ ﾗﾝｽｳ .....
2800    X=R*RND(1)*COS(A)*3+X1 :Y=R*RND(1)*SIN(A)+Y1
2810    GOSUB 3190 :RETURN
2820 '
```

- 2580: 爆発半径の指定（2800行で使用）
- 2600–2620: 音出し
- 2660: ランダムに赤い線を引く
- 2700: 1回目は赤，2回目は黄色で半径が37
- 2710–2730: ランダムに点を打つ
- 2760: 時間待ち
- 2800: ランダムな点をえらぶ
- 2810: 音出し

ドクロ島の崩壊サブルーチン

```
2830 REM ------- ﾎｳｶｲ -------
2840  COLOR C2                                        色を赤に指定
2850  FOR I=23 TO 5 STEP -3
2860   FOR XILND=23 TO I STEP -3
2870    GOSUB 3190                                    音出しサブルーチン
2880    GET@ (XILND,6)-(XILND+2,21),A                 ドクロ島の読み込み
2890    PUT@ (XILND,7)-(XILND+2,22),A                 1段下げを書き出し
2900   NEXT XILND
2910  NEXT I
2920 '
2930  FOR YILND=8 TO 23
2940   GOSUB 3190                                     音出しサブルーチン
2950   GET@ (0,YILND)-(24,23),A                       ドクロ島の読み込み
2960   GOSUB 3190                                     音出しサブルーチン
2970   PUT@ (0,YILND+1)-(24,24),A                     1段下げて書き出し
2980   GOSUB 3190                                     音出しサブルーチン
2990  NEXT YILND
3000 '
3010  FOR I=1 TO 50 }
3020   GOSUB 3190   }                                 音出しサブルーチン
3030  NEXT I
3040 '
3050  FOR I=1 TO 100 :NEXT I                          時間つぶし
3060  RETURN 980
3070 '
```

その他のサブルーチン

```
3080 REM ----- ｶｳﾝﾀｰ -----
3090  COLOR C7
3100  TM=TM-1
3110  LOCATE 65,1 :PRINT USING "###";TM;
3120  IF TM<>0 THEN RETURN
3130   BEEP
3140   COLOR C6 :LOCATE 35,4 :PRINT "ｼﾞｶﾝｷﾞﾚ !!!"
3150   FOR K=1 TO 20 :BEEP 1 :FOR J=1 TO 13 :BEEP 0 :NEXT J :NEXT K    音出し
3160   GOSUB 3940
3170   RETURN 1290
3180 '
3190 REM ------- BEEP -------
3200  BEEP1                   }
3210   FOR J=1 TO 7 :NEXT J   }                       音出しサブルーチン
3220  BEEP0                   }
3230  RETURN
3240 '
3250 REM ----- Inkey ﾙｰﾁﾝ -----
3260  I$=INKEY$                        }
3270  I$=INKEY$ :IF I$="" THEN 3270    }              キー入力待ち
3280  RETURN                           }
3290 '
```

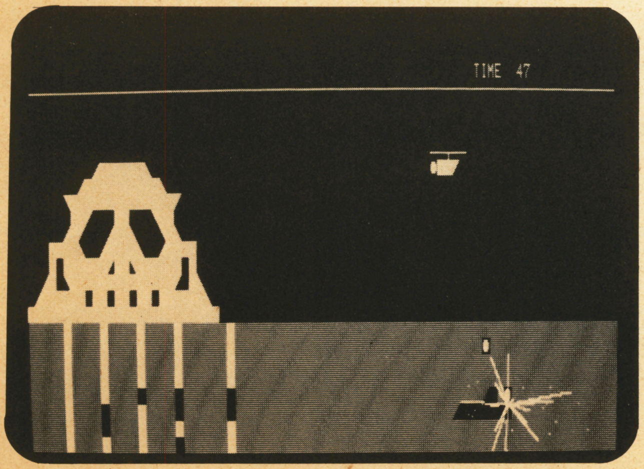
遊び方の説明

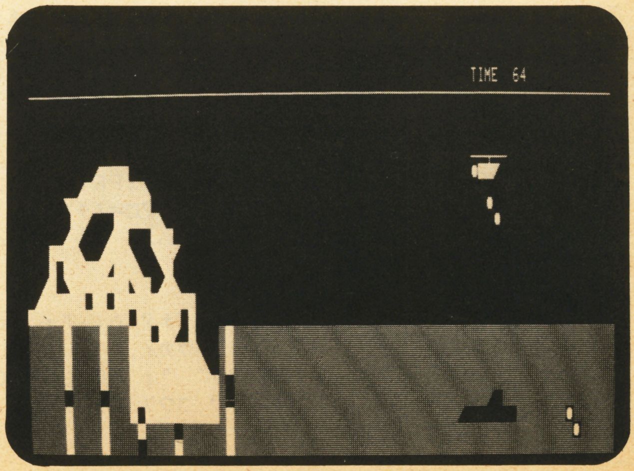
```
3300 REM ----- ｱｿﾋﾞｶﾀ ﾉ ｾﾂﾒｲ -----
3310  BEEP :LOCATE 20,8 :PRINT "ｱｿﾋﾞｶﾀ ﾉ ｾﾂﾒｲ ｶﾞ ﾋﾂﾖｳ ﾃﾞｽｶ (Y/N) ?";
3320  GOSUB 3250 :ANS$=I$ :PRINT ANS$
3330  IF ( ANS$<>"Y" ) AND ( ANS$<>"y" ) THEN RETURN
3340   CLS :COLOR C5 :LOCATE 25,2 :PRINT "***"   説明が必要なければすぐサブルーチンからぬける
3350   LOCATE 45,2 :PRINT "***"
3360   COLOR C6 :LOCATE 30,2 :PRINT "ｱｿﾋﾞｶﾀ ﾉ ｾﾂﾒｲ";
3370 '
3380   COLOR C7 :LOCATE 14,6 :PRINT "ｺﾉ ｹﾞｰﾑ ﾊ, ｷﾒﾗﾚﾀ ｼﾞｶﾝ ﾅｲ ﾆ";
3390   COLOR C2 :PRINT " 'ﾄﾞｸﾛ=ｷﾁ'";
3400   COLOR C7 :PRINT " ｦ ﾊｶｲ ｽﾙ ｹﾞｰﾑ ﾃﾞｽ｡"
3410   GOSUB 3940
3420 '
3430   LOCATE 7,9 :PRINT "ｱﾅﾀ ﾊ, "; :COLOR C5 :PRINT "'ｸﾛ ﾉ ｾﾝｽｲｶﾝ' ";
3440   COLOR C7 :PRINT "ｦ ﾂｶｲ, "; :COLOR C6 :PRINT "'ﾋﾞｰﾑ ﾎｳ' ";
3450   COLOR C7 :PRINT "ﾃﾞ "; :COLOR C2 :PRINT "'ﾄﾞｸﾛ=ﾄｳ' ﾉ ｱｼ ";
3460   COLOR C7 :PRINT "ｦ ｺｳｹﾞｷ ｼﾏｽ｡"
3470   GOSUB 3940
3480 '
```

```
3490  LOCATE 7,11 :COLOR C2 :PRINT "'ド゛クロ=トウ' ノ アシ";
3500  COLOR C7 :PRINT " ヲ, 4 ホン トモ ";
3510  COLOR C6 :PRINT "'ビ゛ーム ホウ'"; :COLOR C7 :PRINT " デ゛ ハカイスルト, ";
3520  COLOR C3 :PRINT "アナタ ノ カチ "; :COLOR C7 :PRINT "デ゛ス。"
3530  GOSUB 3940
3540 '
3550  LOCATE 7,14 :PRINT "テキ ハ, "; :COLOR C4 :PRINT "'ヘリコプ゜ター'";
3560  COLOR C7 :PRINT " カラ ";
3570  COLOR C6 :PRINT "'バ゛クダ゛ン' "; :COLOR C7 :PRINT "ヲ オトシテ, コウゲ゛キ シテ キマス。"
3580  GOSUB 3940
3590 '
3600  LOCATE 13,17 :PRINT "ウマク カワシナガ゛ラ, テキ ヲ コウゲ゛キ シテ クダ゛サイ。"
3610  GOSUB 3940
3620 '
3630  COLOR C3 :LOCATE 20,21 :PRINT "ツギ゛ニ, Key ノ セツメイ ヲ シマス。"
3640  COLOR C2 :LOCATE 65,23 :PRINT "Hit Any Key !" :GOSUB 3250 ──── キー入力待ち
3650 '
3660 REM   Key ノ セツメイ
3670  CLS :COLOR C4 :LOCATE 25,2 :PRINT "***"
3680  LOCATE 45,2 :PRINT "***"
3690  COLOR C6 :LOCATE 30,2 :PRINT "Key ノ セツメイ";
3700 '
3710  COLOR C6 :LOCATE 18,8 :PRINT "'ビ゛ーム ホウ'"
3720  COLOR C5 :LOCATE 51,7 :PRINT "'センスイカン'"; :COLOR C7 :PRINT " ヲ ウゴ゛カス"
3730 '
3740  COLOR C6 :LOCATE 15,11 :PRINT "████████████████"
3750  COLOR C3 :LOCATE 15,13 :PRINT "( スペ゜ース  バ゛ー )"
3760 '
3770  COLOR C3 :LOCATE 45,10 :PRINT "             ( ウエ )"
3780  COLOR C7 :LOCATE 45,12 :PRINT "                8"
3790  COLOR C2 :LOCATE 45,14 :PRINT "                I"
3800  COLOR C2 :LOCATE 45,15 :PRINT "                I"
3810 '
3820  COLOR C3 :LOCATE 43,16 :PRINT "( ヒダ゛リ ) "; :COLOR C7 :PRINT " 4 ";
3830  COLOR C2 :PRINT " ---+--- "; :COLOR C7 :PRINT " 6 ";
3840  COLOR C3 :PRINT " ( ミギ゛ )"
3850 '
3860  COLOR C2 :LOCATE 45,17 :PRINT "                I"
3870  COLOR C2 :LOCATE 45,18 :PRINT "                I"
3880  COLOR C7 :LOCATE 45,20 :PRINT "                2"
3890  COLOR C3 :LOCATE 45,22 :PRINT "             ( シタ )"
3900 '
3910  COLOR C4 :LOCATE 60,24 :PRINT "Hit Any Key !"; :GOSUB 3250 ──── キー入力待ち
3920  RETURN
3930 '
```

時間待ちサブルーチンとカラーデータ

```
3940 REM ----- ジ゛カン マチ ループ゜ -----
3950  FOR I= 1 TO 2300
3960  NEXT I
3970  RETURN
3980 '
3990 REM ----- カラー デ゛ータ -----
4000  DATA  0, 1, 2, 3, 4, 5, 6, 7
```





(NEC) PC8001への移植のPoint

PC のプログラムは，ほぼ FM と同じだが，SCREEN関数の代わりにPEEKサブルーチンを追加する必要がある。

CLS はPRINT CHR$(12) で表す。

CONSOLE W, X, Y, Z 第4パラメータのカラーモードを，FM は0を，PC は1を指定する。

COLOR X, Y, Z PCではプログラムの最初に出てくるCOLORの第3パラメータで，キャラクタおよびグラフィックのモード指定が必要。今回のプログラムではキャラクタを主に使用するので0を指定した。

LINE グラフィックのドットの大きさが FM と PC では違うので，X座標，Y座標を調整する必要がある。

RANDOMIZE PC にはないので削除する。

INKEY $ PC ではINKEY$がうまく使えないのでINP関数を代用する。

SCREEN PC にはSCREEN関数がないので，V-RAMを直接PEEKするサブルーチンを追加した。

```
100 PRINT CHR$(12) :WIDTH 80,25 :DEFINT B-Z :CONSOLE 0,25,0,1
180  COLOR C7,,0 :LOCATE 32,2 :PRINT "Mission III :"
405  COLOR C5
410  LINE (83/4,6)-(527/4,193-100),PSET,B
420  LINE (75/4,3)-(535/4,196-100),PSET,B
430  LINE ((70-3)/4,1)-(540/4,198-100),PSET,B
470 FOR I=1 TO 1500:NEXT I:PRINT CHR$(12):GOSUB 1570:LOCATE 10,5:PRINT "ドクロ=トウ"
960  COLOR C7 :PRINT CHR$(12) :GOSUB 3300
1020  TM=TM+100
1065  COLOR C7
1070  LINE (0,20/2)-(639/4,20/2),PSET
1150 ''サクジョ
1160   IF INP(9)=191 THEN GOSUB 1870 :IF XBEEM=40/4 THEN GOSUB 2830 ELSE 1220
1170   IF INP(0)=251 AND YSBM<23 THEN YSBM=YSBM+1 :GOSUB 1800 :GOTO 1220
1180   IF INP(1)=254 AND YSBM>17 THEN YSBM=YSBM-1 :GOSUB 1800 :GOTO 1220
1190   IF INP(0)=191 AND XSBM<70 THEN XSBM=XSBM+1 :GOSUB 1800 :GOTO 1220
1200   IF INP(0)=239 AND XSBM>40 THEN XSBM=XSBM-1 :GOSUB 1800 :GOTO 1220
1210   IF INP(1)=191 THEN GOSUB 2060:' ワープ
1270 GOTO 1160
1300  COLOR C6 :PRINT CHR$(12) :K=12 :LOCATE 0,4
1500  COLOR C7 :PRINT CHR$(12)
1595  COLOR C2
1755  COLOR C6
1880  XBEEM=XWLL*8/4
1890  YBEEM=YSBM*8/2
1905  GOSUB 5000 :' WALL CHECK
1910  IF NOT WCHK THEN XBEEM=XBEEM-56/4 :GOTO 1965 ELSE XBEEM=XBEEM+8/4
1915   COLOR C6
1920   LINE (XSBM+3,YBEEM/4)-(XBEEM/2,YBEEM/4),"-"
1935   COLOR C1
1940   LINE (XSBM+3,YBEEM/4)-(XBEEM/2,YBEEM/4),"■"
1965   GOSUB 6000 :'FOOT CHECK
1970   IF NOT FCHK THEN XBEEM=XBEEM-40/4 :GOTO 1965
1975    COLOR C6
1980    LINE (XSBM+3,YBEEM/4)-(XBEEM/2,YBEEM/4),"-"
1995    COLOR C1
2000    LINE (XSBM+3,YBEEM/4)-(XBEEM/2,YBEEM/4),"■"
2005    COLOR C0
2010    FOR I=20 TO XBEEM/8*4 STEP -5
2020     PUT@ (XSBM,YSBM)-(XSBM+7,YSBM+1),SMOLD,C1
2535  GOSUB 7000 :'BOMB CHECK
2540  IF NOT BCHK THEN 2450
2550  X1=( XB(I)*8+4 )/4 :Y1=( YB(I)*8+8 )/2
2580   R=30/3
2645   COLOR C2
2700   CC=C2 :IF K=2 THEN CC=C6 :R=37/3
2800    X=R*RND(1)*COS(A)*1.6+X1 :Y=R*RND(1)*SIN(A)+Y1
3060  GOTO 980
3170   GOTO 1290
3340   PRINT CHR$(12) :COLOR C5 :LOCATE 25,2 :PRINT "***"
3670   PRINT CHR$(12) :COLOR C4 :LOCATE 25,2 :PRINT "***"
5000 REM ------ WALL CHECK ------
5010  WCHK=TRUE
5020  J=PEEK ( &HF301+XWLL+YSBM*120 )
5030  IF CHR$(J)=" " THEN WCHK=FALSE
5040  RETURN
5050 '
6000 REM ------ FOOT CHECK ------
6010  FCHK=TRUE
6020  J=PEEK ( &HF301+XBEEM/2+YSBM*120 )
6030  IF CHR$(J)=" " THEN FCHK=FALSE
6040  RETURN
6050 '
7000 REM ------ BOMB CHECK ------
7010  BCHK=FALSE
7020  J=PEEK ( &HF301+XB(I)+( YB(I)+1 )*120 )
7030  IF ( CHR$(J)="◤" ) OR ( CHR$(J)=" " ) THEN BCHK=TRUE
7040  RETURN
```

(日立) LEVEL 3への移植のPoint

L 3 にはGET @・PUT @がないので，絵の移動は，キャラクタを配列に読み込み，LOCATEで位置を指定してPRINTで表示する。そのため，ドクロ島のデータと表示の部分をサブルーチン化して追加する必要がある。

また，L 3 の処理速度は FM より遅いので時間待ちのFOR～NEXTを使うときはループの回数を減らす。

NEW ON 80字モードで処理するため，RUNする前にNEW ON & H17かNEW ON 7と入力する必要がある。

WIDTH L 3 には第2パラメータがない。

CONSOLE W, X, Y, Z 第4パラメータがない。

SCREEN文 L 3 では画面のモードをプログラムの最初でSCREEN文を使って設定する必要がある。第1パラメータでハイレゾリューションモードを，第3パラメータでキャラクタを表示するノンインターレースモードを指定する。

BEEP 1・BEEP 0 BEEPの長さを指定するスイッチがないのでBEEPで代用する。

GET @・PUT @ L 3 には，GET @・PUT @がないので，LOCATEとPRINTを使う。

```
100 WIDTH 80 :DEFINT B-Z :CONSOLE 0,25,0 :SCREEN 1,,1
110 DIM  FOOT$(7,2) ,PNT(2) ,DK$(9,3)
120 ''サクジョ
460  YHEL=4 :NBMB=4 :XWLL=27 :YWLL=22 :FALSE=0 :TRUE=NOT FALSE :GOSUB 5000
470  FOR I=1 TO 500 :NEXT I :CLS :GOSUB 1570 :LOCATE 10,5 :PRINT "ドクロ=トウ"
530   LOCATE 50,8 :RHEL$="  ■■● " :PRINT RHEL$
540 ''540->600, 640->680, 730, 780->810, 860, 920, 1700->1740 サクジョ
910   LOCATE 50,8 :LHEL$=" ●■■  " :PRINT  LHEL$
1035  FOR I=0 TO 2 :PNT(I)=0 :FOR J=0 TO 7 :FOOT$(J,I)="■" :NEXT J: NEXT I
1110  FOR I=1 TO 5 :BEEP :FOR J=1 TO 50 :NEXT J :NEXT I
1640  FOR I=9 TO 0 STEP -1
1650   LOCATE 0,7+I
1660   FOR J=0 TO 3
1670    PRINT DK$(I,J);
1680   NEXT J
1690  NEXT I
1810  COLOR C1
1815  LOCATE XOLDSBM,YOLDSBM   :PRINT "■■■■■■■■";
1816  LOCATE XOLDSBM,YOLDSBM+1 :PRINT "■■■■■■■■";
1820
1830  LOCATE XSBM,YSBM   :PRINT "■■■◤ ■■";
1835  LOCATE XSBM,YSBM+1 :PRINT "◤      ";
2020     LOCATE I,YBEEM/8 :PRINT " ";
2025     IF I>5 THEN FOOT$( (PNT(I/5-2)+YSBM-17) MOD 8,I/5-2 )=" "
2080   WARP=FALSE :COLOR C1
2090   LOCATE XSBM,YSBM   :PRINT "■■■■■■■■";
2095   LOCATE XSBM,YSBM+1 :PRINT "■■■■■■■■";
2152  COLOR C6
2153  LOCATE XWLL,YWLL-1 :PRINT "■";
2154  LOCATE XWLL,YWLL   :PRINT " ";
2155  LOCATE XWLL,YWLL+1 :PRINT " ";
2160  LOCATE XWLL,YWLL+2 :PRINT "■";
2200  FTILND=INT(RND(1)*3)
2205 '
2206  COLOR C2
2210  PNT(FTILND)=PNT(FTILND)-1 :IF PNT(FTILND)<0 THEN PNT(FTILND)=7
2216  FOOT$( PNT(FTILND),FTILND )="■"
2220  FOR K=17 TO 24
2225   LOCATE FTILND*5+10,K :PRINT FOOT$( PNT(FTILND),FTILND );
2227   PNT(FTILND)=( PNT(FTILND)+1 ) MOD 8
2228  NEXT K
2280   COLOR C4 :LOCATE XHEL,YHEL :PRINT " ─┬─ "
2290   LOCATE XHEL,YHEL+1 :IF DR=1 THEN PRINT RHEL$ ELSE PRINT LHEL$
2450    LOCATE XB(I),YB(I) :PRINT "■";
2480     COLOR C2 :LOCATE XB(I),YB(I) :PRINT "●";
2590   FOR K=1 TO 5
2600    BEEP
2610    FOR J=1 TO 25 :NEXT J
2620 ''2620, 3220 サクジョ
2850  XILND=20 :YILND=9 :DKR=3 :GOSUB 6500 :GOSUB 6000 :'RIGHT 5 BLOCKS
2860  '
2870  YILND=11 :GOSUB 6500 :GOSUB 6000 :'ALSO,RIGHT 5 BLOCKS
2880  XILND=15 :YILND=9 :DKR=2 :GOSUB 6500 :GOSUB 6000 :' NEXT RIGHT 5 BLOCKS
2890  '
2900  XILND=20 :YILND=13 :DKR=3 :GOSUB 6500 :GOSUB 6000 :' RIGHT 5 BLOCKS
2910  XILND=15 :YILND=11 :DKR=2 :GOSUB 6500 :GOSUB 6000 :' NEXT RIGHT 5 BLOCKS
2920  XILND=10 :YILND=9 :DKR=1 :GOSUB 6500 :GOSUB 6000 :' NEXT LEFT 5 BLOCKS
2930  '
2940  FOR I=8 TO 24 STEP 2
2950   XILND=20 :YILND=I+6 :DKR=3 :GOSUB 6500 :GOSUB 6000
2960   XILND=15 :YILND=I+4 :DKR=2 :GOSUB 6500 :GOSUB 6000
2970   XILND=10 :YILND=I+2 :DKR=1 :GOSUB 6500 :GOSUB 6000
2980   XILND=0 :YILND=I   :DKR=0 :GOSUB 6000
2985   XILND=7 :GOSUB 6500: PRINT "   ";
2990  NEXT I
2995  LOCATE 0,24 :PRINT SPACE$(26);
3010  FOR I=1 TO 25
3150   FOR K=1 TO 10 :BEEP :FOR J=1 TO 50 :NEXT J :NEXT K
3200  BEEP
3260  FOR I=0 TO 30 :I$=INKEY$ :NEXT I
4000 DATA  0, 1, 2, 3, 4, 5, 6, 7
5000 '
5010 REM ------ ドクロ=トウ ノ テイギ ------
5020  FOR I=0 TO 9
5030   FOR J=0 TO 3
5040    READ DK$(I,J)
5050   NEXT J
5060  NEXT I
5120  RETURN
6000 '
6010 REM ----- ホウカイ Subroutine 2 -----
6020  FOR K=9 TO 0 STEP -1
6030   IF ( YILND+K )>24 THEN 6060
6040    LOCATE XILND,YILND+K
6050    PRINT DK$(K,DKR);
6060  NEXT K
6070  GOSUB 3190
6080 RETURN
6090 '
6500 REM ----- クウハク ヲ カク -----
6510 FOR K=1 TO 2
6520  IF ( YILND-K )>24 THEN 6550
6530   LOCATE XILND,YILND-K
6540   PRINT SPACE$(5);
6550 NEXT K
6560 RETURN
6570 '
7000 REM ------ ドクロ=トウ ノ データ ノ テイギ ------
7010 DATA "    ◢","■■■■","◣   ","     "
7020 DATA "   ■■","■■■■","■■  ","     "
7030 DATA "  ■■■","■■■■","■■■ ","     "
7040 DATA "  ■  ","  ■  ","  ■■ ","     "
7050 DATA "  ■  ","  ■■ ","  ■■ ","     "
7060 DATA " ■■ ◢","■■■■","  ■■","■    "
7070 DATA " ■■■■","■■■■","■■■■"," ■   "
7080 DATA " ■■■■","■■■■","■■■■"," ■■  "
7090 DATA "■■■ ■","■ ■■","■ ■■","■■■  "
7100 DATA "■■■■■","■■■■","■■■■","■■■■■"
```

★これらの機種で遊ぶにはFM8用のものを上記の行だけ変更してください。

(シャープ)MZ700への移植のPoint

基本的には [L3] への移植のポイントと同じだが，[MZ] には，1行40字モードしかないのでドクロ島・潜水艇・対潜ヘリを小さくし，その他のX座標（一部はY座標も）も調整する必要がある。タイトル・ゲームオーバーの文字の太さやゲームの説明部分の配置も変更した。

また，[MZ] にはテンキーがないので潜水艇の操縦にはカーソル移動キーを使う。

[MZ] の処理速度は [FM] より早いので時間待ちのFOR～NEXTを使うときはループの回数を増やす。

WIDTH　[MZ] の画面は40字×25行に固定されている。

CONSOLE W, X, Y, Z　[MZ] の第3パラメータは横スクロールの開始を，第4パラメータはスクロール字数を設定する。

RANDOMIZE　[MZ] にはないので削除する。

LINE　[MZ] にはLINEがないので，PSETとFOR～NEXTあるいはPRINTとFOR～NEXTを使って処理する。

SCRN$　[MZ] のSCRN関数は [FM] のSCREEN関数とCHR$関数を合わせたものである。

GET @・PUT @　[MZ] には，GET @・PUT @がないので，LOCATEとPRINTを使う。

PAI（　）　[MZ] は最初からπという関数をもっているので，それを利用する。

```
100 CLS :DEFINT B-Z :CONSOLE 0,25,0,40
110 DIM  FOOT$(7,2) ,PNT(2) ,DK$(9,3)
120 ''120, 410->440, 540->600, 640->680, 730, 780->810 860, 920, 1700->1740 ｻｸｼﾞｮ
180  COLOR C7 :LOCATE 13,1 :PRINT "Mission III :"
190  COLOR C1 :LOCATE 0,3 :K=4
210  PRINT TAB(K);"■ ■■ ◢■■ ■■■ ■   ■    "
220  PRINT TAB(K);"■  ／ ■ ■ ■■■ ■■■ ■■■"
230  PRINT TAB(K);"◣  ／ ■ ■ ■   ◣  ／"
240  PRINT TAB(K);"■ ＼ ／ ■ ■■■ ■ ＼  ／"
250  PRINT TAB(K);"■  ／  ■■■   ■   ／"
270  PRINT :COLOR C6 :K=8
280  PRINT TAB(K);"       ■   ／"
290  PRINT TAB(K);"     ■■■■ ／ "
300  PRINT TAB(K);" ／ ＼ ■ ■ ◢■ "
310  PRINT TAB(K);" ／  ＼ ／／ ■ "
320  PRINT TAB(K);"／    ＼／ ／  ■ "
340  PRINT :COLOR C3 :PRINT :K=5
350  PRINT TAB(K);" ■ ■ ◢■■ ■         "
360  PRINT TAB(K);"■■■■ ／／ ■■■◤ ■■ "
370  PRINT TAB(K);" ■ ■  ／  ■◤    ／"
380  PRINT TAB(K);"   ／ ／   ■    ／ "
390  PRINT TAB(K);"  ／ ／    ■■■ ■◤ "
460  YHEL=4 :NBMB=4 :XWLL=15 :YWLL=22 :FALSE=0 :TRUE=NOT FALSE :GOSUB 5000
470  GOSUB 3940 :CLS :GOSUB 1570 :LOCATE 4,5 :PRINT "ﾄﾞｸﾛ=ﾄｳ"
510   COLOR C4 :LOCATE 25,5 :PRINT "ﾍﾘｺﾌﾟﾀｰ"
520   LOCATE 25,7 :PRINT " ─┬─ "
530   LOCATE 25,8 :RHEL$=" ◥■● " :PRINT RHEL$
630   LOCATE 26,10 :PRINT "●";" ﾊﾞｸﾀﾞﾝ"
700   LOCATE 30,19 :PRINT "ｾﾝｽｲｶﾝ" :COLOR C1
710   LOCATE 30,21 :PRINT "■◤ ■";
720   LOCATE 30,22 :PRINT "◤    ";
760   COLOR C6 :LOCATE XWLL+1,YWLL :PRINT "ﾎﾞｳｷﾞｮｾﾝ"
770   GOSUB 2120
840   COLOR C2 :LOCATE 12,19 :PRINT "ｱｼ"
850   LOCATE 11,20 :PRINT " ";
900   LOCATE 25,7 :PRINT " ─┬─ "
910   LOCATE 25,8 :LHEL$=" ●■◤ " :PRINT  LHEL$
990  XHEL=25 :DR=1
1010  XSBM=30 :YSBM=21 :XOLDSBM=30 :YOLDSBM=21 :WARP=TRUE
1020  TM=TM+100
1035  FOR I=0 TO 2 :PNT(I)=0 :FOR J=0 TO 7 :FOOT$(J,I)="■" :NEXT J: NEXT I
1070  FOR I=0 TO 79 :PSET (I,5,C7) :NEXT I
1080  COLOR C7 :LOCATE 30,1 :PRINT "TIME";TM
1100  COLOR C6 :LOCATE 20,10 :PRINT "START"
1110  FOR I=1 TO 30 :BEEP 1 :FOR J=1 TO 40 :NEXT J:BEEP 0 :NEXT I
1120  COLOR C0 :LOCATE 20,10 :PRINT "START" :COLOR C7
1150  I$=INKEY$(0)
1160   IF I$=" " THEN GOSUB 1870 :IF XBEEM=4 THEN GOSUB 2830 ELSE 1220
1170   IF I$="↓" AND YSBM<23 THEN YSBM=YSBM+1 :GOSUB 1800 :GOTO 1220
1180   IF I$="↑" AND YSBM>17 THEN YSBM=YSBM-1 :GOSUB 1800 :GOTO 1220
1190   IF I$="→" AND XSBM<34 THEN XSBM=XSBM+1 :GOSUB 1800 :GOTO 1220
1200   IF I$="←" AND XSBM>19 THEN XSBM=XSBM-1 :GOSUB 1800 :GOTO 1220
1300  COLOR C6,C2 :CLS :K=3 :LOCATE 0,4
1320  PRINT TAB(K);" ◢■■◣   ▲   ◣  ◢ ■■■■■"
1330  PRINT TAB(K);"◢    ＼ ◢ ◣  ■◣◢■ ■    "
1340  PRINT TAB(K);"■      ／   ＼ ■ ◥ ■ ■    "
1350  PRINT TAB(K);"■     ／     ＼■ ▼ ■ ■■■■ "
1360  PRINT TAB(K);"■  ■■ ■■■■■■■ ■   ■ ■    "
1370  PRINT TAB(K);"◥    ／ ■     ■ ■   ■ ■    "
1380  PRINT TAB(K);" ◥■■◤  ■     ■ ■   ■ ■■■■■"
1410  PRINT TAB(K);" ◢■■◣ ■     ■ ■■■■■ ■■■■◣ "
1420  PRINT TAB(K);"◢    ◣ ■     ■ ■     ■    ◣"
1430  PRINT TAB(K);"■    ■ ■     ■ ■     ■   ◢"
1440  PRINT TAB(K);"■    ■ ＼   ／ ■■■■  ■■■◤ "
1450  PRINT TAB(K);"■    ■  ＼ ／  ■     ■  ＼ "
1460  PRINT TAB(K);"◥    ◤   ＼／  ■     ■   ＼"
1470  PRINT TAB(K);" ◥■■◤     ▼   ■■■■■ ■    ◥"
1490  GOSUB 3940
1540   IF ANS$="Y" THEN YHEL=4: TM=0: GOTO 980
1580  COLOR C1
1590  FOR I=17 TO 24 :FOR J=0 TO 38
1593   LOCATE J,I :PRINT "■";
1596  NEXT J:NEXT I
1600  COLOR C2 :FOR I=0 TO 3
1610   FOR J=17 TO 24 :LOCATE I*3+2,J :PRINT "■"; :NEXT J
1620  NEXT I
1640  FOR I=9 TO 0 STEP -1
1650   LOCATE 1,7+I
1660   FOR J=0 TO 3
1670    PRINT DK$(I,J);
1680   NEXT J
1690  NEXT I
1760  COLOR C6 :FOR I=17 TO 24 :LOCATE XWLL,I :PRINT "■"; :NEXT I
1810  COLOR C1
1815  LOCATE XOLDSBM,YOLDSBM   :PRINT "■■■■";
1816  LOCATE XOLDSBM,YOLDSBM+1 :PRINT "■■■■";
1830  LOCATE XSBM,YSBM   :PRINT "■◤ ■";
1835  LOCATE XSBM,YSBM+1 :PRINT "◤   ";
1880  XBEEM=XWLL*2
1890  YBEEM=YSBM*2
1910  IF SCRN$( XWLL,YSBM,1 )=" " THEN XBEEM=XBEEM-8 :GOTO 1970
1920   COLOR C6:FOR I=XSBM+1 TO XWLL+1 STEP -1:LOCATE I,YSBM:PRINT "-";:NEXT I
1940   COLOR C1:FOR I=XSBM+1 TO XWLL+1 STEP -1:LOCATE I,YSBM:PRINT "■";:NEXT I
1970   IF SCRN$( XBEEM/2,YSBM,1 )=" " THEN XBEEM=XBEEM-6 :GOTO 1970
1980   COLOR C6:FOR I=XSBM+1 TO XBEEM/2+1 STEP-1:LOCATE I,YSBM:PRINT "-";:NEXT I
2000   COLOR C1:FOR I=XSBM+1 TO XBEEM/2+1 STEP-1:LOCATE I,YSBM:PRINT "■";:NEXT I
2010    FOR I=11 TO XBEEM/2 STEP -3
2020     LOCATE I,YBEEM/2 :PRINT " ";
2025     IF I>3 THEN FOOT$( (PNT(I/3-2)+YSBM-17) MOD 8,I/3-2 )=" "
2080   WARP=FALSE :COLOR C1
2090   LOCATE XSBM,YSBM   :PRINT "■■■■";
2095   LOCATE XSBM,YSBM+1 :PRINT "■■■■";
2152  COLOR C6
2153  LOCATE XWLL,YWLL-1 :PRINT "■";
2154  LOCATE XWLL,YWLL   :PRINT " ";
2155  LOCATE XWLL,YWLL+1 :PRINT " ";
2160  LOCATE XWLL,YWLL+2 :PRINT "■";
2200  FTILND=INT(RND(1)*3)
2205 '
2206  COLOR C2
2210  PNT(FTILND)=PNT(FTILND)-1 :IF PNT(FTILND)<0 THEN PNT(FTILND)=7
2216  FOOT$( PNT(FTILND),FTILND )="■"
2220  FOR K=17 TO 24
2225   LOCATE FTILND*3+5,K :PRINT FOOT$( PNT(FTILND),FTILND );
2227   PNT(FTILND)=( PNT(FTILND)+1 ) MOD 8
2228  NEXT K
2280   COLOR C4 :LOCATE XHEL,YHEL :PRINT " ─┬─ "
2290   LOCATE XHEL,YHEL+1 :IF DR=1 THEN PRINT RHEL$ ELSE PRINT LHEL$
2310    IF XHEL>34 THEN DR=-1 :GOTO 2340
2320    IF XHEL<15 THEN DR=1
2450    LOCATE XB(I),YB(I) :PRINT "■";
2480     COLOR C2 :LOCATE XB(I),YB(I) :PRINT "●";
2540  IF SCRN$( XB(I),YB(I)+1,1 )="■" THEN 2450
2550  X1=XB(I)*2+1 :Y1=YB(I)*2+2
2580   R=6
2650   FOR A=0 TO PAI(2) STEP .4
2660    GOSUB 2790 :PSET (X,Y,C6)
2700    CC=C2 :IF K=2 THEN CC=C6 :R=10
2710    FOR A=0 TO PAI(2) STEP .5
2760  FOR I=1 TO 4000 :NEXT I
2800    X=R*RND(1)*COS(A)*2+X1 :IF X>79 THEN 2800
2805    Y=R*RND(1)*SIN(A)+Y1 :IF Y>49 THEN 2805
2850  XILND=10 :YILND=9 :DKR=3 :GOSUB 6500 :GOSUB 6000 :'RIGHT 5 BLOCKS
2860  '
2870  YILND=11 :GOSUB 6500 :GOSUB 6000 :'ALSO,RIGHT 5 BLOCKS
2880  XILND=7 :YILND=9 :DKR=2 :GOSUB 6500 :GOSUB 6000 :' NEXT RIGHT 5 BLOCKS
2890  '
2900  XILND=10 :YILND=13 :DKR=3 :GOSUB 6500 :GOSUB 6000 :' RIGHT 5 BLOCKS
2910  XILND=7 :YILND=11 :DKR=2 :GOSUB 6500 :GOSUB 6000 :' NEXT RIGHT 5 BLOCKS
2920  XILND=4 :YILND=9 :DKR=1 :GOSUB 6500 :GOSUB 6000 :' NEXT LEFT 5 BLOCKS
2930  '
2940  FOR I=8 TO 24 STEP 2
2950   XILND=10 :YILND=I+6 :DKR=3 :GOSUB 6500 :GOSUB 6000
2960   XILND=7 :YILND=I+4 :DKR=2 :GOSUB 6500 :GOSUB 6000
2970   XILND=4 :YILND=I+2 :DKR=1 :GOSUB 6500 :GOSUB 6000
2980   XILND=0 :YILND=I   :DKR=0 :GOSUB 6000
2985   XILND=3 :GOSUB 6500: PRINT "    ";
2990  NEXT I
2995  LOCATE 0,24 :PRINT SPACE$(12);
3140   COLOR C6 :LOCATE 17,4 :PRINT "ｼﾞｶﾝｷﾞﾚ !!!"
3150   FOR K=1 TO 20 :BEEP 1 :FOR J=1 TO 40 :NEXT J:BEEP 0 :NEXT K
3270  I$=INKEY$(1) :IF I$="" THEN 3270
3340   CLS :COLOR C5 :LOCATE 7,2 :PRINT "***"
3350   LOCATE 25,2 :PRINT "***"
3360   COLOR C6 :LOCATE 11,2 :PRINT "ｱｿﾋﾞｶﾀ ﾉ ｾﾂﾒｲ";
3380   COLOR C7 :LOCATE 2,5 :PRINT "ｺﾉ ｹﾞｰﾑ ﾊ, ｷﾒﾗﾚﾀ ｼﾞｶﾝ ﾅｲ ﾆ";
3430   LOCATE 2,8 :PRINT "ｱﾅﾀ ﾊ, "; :COLOR C5 :PRINT "'ｸﾛｲ ｾﾝｽｲｶﾝ' ";
3490   LOCATE 1,11 :COLOR C2 :PRINT "'ﾄﾞｸﾛ=ﾄｳ' ﾉ ｱｼ";
3550   LOCATE 2,14 :PRINT "ﾃｷ ﾊ, "; :COLOR C4 :PRINT "'ﾍﾘｺﾌﾟﾀｰ'";
3600   LOCATE 2,18 :PRINT "ｳﾏｸ ｶﾜｼﾅｶﾞﾗ, ﾃｷ ｦ ｺｳｹﾞｷ ｼﾃ ｸﾀﾞｻｲ｡"
3630   COLOR C3 :LOCATE 10,21 :PRINT "ﾂｷﾞﾆ, Key ﾉ ｾﾂﾒｲ ｦ ｼﾏｽ｡"
3640   COLOR C2 :LOCATE 25,23 :PRINT "Hit Any Key !" :GOSUB 3250
3670   CLS :COLOR C4 :LOCATE 8,2 :PRINT "***"
3680   LOCATE 23,2 :PRINT "***"
3690   COLOR C6 :LOCATE 12,2 :PRINT "Key ﾉ ｾﾂﾒｲ";
3710   COLOR C6 :LOCATE 3,8 :PRINT "'ﾋﾞｰﾑ ﾎｳ'"
3720   COLOR C5 :LOCATE 18,7 :PRINT "'ｾﾝｽｲｶﾝ'"; :COLOR C7 :PRINT "ｦ ｳｺﾞｶｽ"
3740   COLOR C6 :LOCATE 0,11 :PRINT "■■■■■■■■■■"
3750   COLOR C3 :LOCATE 0,13 :PRINT "( ｽﾍﾟｰｽ  ﾊﾞｰ )"
3770   COLOR C3 :LOCATE 11,10 :PRINT "             ( ｳｴ )"
3780   COLOR C7 :LOCATE 11,12 :PRINT "               ↑"
3790   COLOR C2 :LOCATE 11,14 :PRINT "               I"
3800   COLOR C2 :LOCATE 11,15 :PRINT "               I"
3820   COLOR C3 :LOCATE 11,16 :PRINT "( ﾋﾀﾞﾘ )"; :COLOR C7 :PRINT " ← ";
3830   COLOR C2 :PRINT "---+---"; :COLOR C7 :PRINT " → ";
3840   COLOR C3 :PRINT "( ﾐｷﾞ )"
3860   COLOR C2 :LOCATE 11,17 :PRINT "               I"
3870   COLOR C2 :LOCATE 11,18 :PRINT "               I"
3880   COLOR C7 :LOCATE 11,20 :PRINT "               ↓"
3890   COLOR C3 :LOCATE 11,22 :PRINT "             ( ｼﾀ )"
3910   COLOR C4 :LOCATE 22,24 :PRINT "Hit Any Key !"; :GOSUB 3250
3950  FOR I= 1 TO 6000
5000 '
5010 REM ------ ﾄﾞｸﾛ=ﾄｳ ﾉ ﾃｲｷﾞ ------
5020  FOR I=0 TO 9
5030   FOR J=0 TO 3
5040    READ DK$(I,J)
5050   NEXT J
5060  NEXT I
5120  RETURN
6000 '
6010 REM ----- ﾎｳｶｲ Subroutine 2 -----
6020  FOR K=9 TO 0 STEP -1
6030   IF ( YILND+K )>24 THEN 6070
6040    LOCATE XILND,YILND+K
6050    PRINT DK$(K,DKR);
6060    GOSUB 3190
6070  NEXT K
6080 RETURN
6500 '
6510 REM ----- ｸｳﾊｸ ｦ ｶｸ -----
6520 FOR K=1 TO 2
6530  IF ( YILND-K )>24 THEN 6560
6540   LOCATE XILND,YILND-K
6550   PRINT SPACE$(3);
6560 NEXT K
6570 RETURN
7000 '
7010 REM ------ ﾄﾞｸﾛ=ﾄｳ ﾉ ﾃﾞｰﾀ ﾉ ﾃｲｷﾞ ------
7020 DATA "   ","   ","   ","   "
7030 DATA "  ◢■","◣  ","   ","   "
7040 DATA " ◥■■","■■ ","◤  ","   "
7050 DATA " ／◥","◤  ","＼ ","   "
7060 DATA " ◢ ","|  ","◣ ","   "
7070 DATA " ◣●◢","◣●","◢■","   "
7080 DATA " ◤■o","o■","■◤","   "
7090 DATA "◢ ■■","■■","■ ","◣ "
7100 DATA "◢■ ■ |","| ■"," ■","◣ "
7110 DATA "■■■■","■■","■■","■■"
```


★これらの機種で遊ぶにはFM8用のものを

(東芝)PASOPIAへの移植のPoint

PAS でのCOLORは1行単位で実行される。つまり，新たにCOLORを指定すると，その行の色がすべて変わってしまう。それを避けるために，カラー・アトリビュート・キャラクタをPRINT CHR$（　）で指定することにしたが，アトリビュート文字の分だけキャラクタの位置がずれてしまうので，X座標を調整する必要がある。それ以外は，基本的に L3 への移植と同じだ。

WIDTH　PAS には第2パラメータがない。

SCREEN文　PAS では画面のモードをプログラムの最初でSCREEN文を使って設定する必要がある。SCREEN 2はファイングラフィックの指定だ。

GET @・PUT @　PAS には，キャラクタ単位で読み込んだり，書き出すGET @・PUT @がないので，LOCATEとPRINTを使う。

BEEP 1・BEEP 0　PAS ではSOUND X, Y で音を出す。Xで音の高さを，Yで音の長さを指定する。

```
100 WIDTH 80 :DEFINT B-Z :SCREEN 2 :CLS
110 DIM  FOOT$(7,2) ,PNT(2) ,DK$(9,3)
120 ''120, 540, 620, 640, 660->680, 730, 780->810, 860, 890, 930 ｻｸｼﾞｮ
180  LOCATE 31,2 :PRINT CHR$(C7);"Mission III :"
190  LOCATE 0,5 :K=13
210  PRINT TAB(K);CHR$(C1);"                     "
220  PRINT TAB(K);CHR$(C1);"                     "
230  PRINT TAB(K);CHR$(C1);"                     "
240  PRINT TAB(K);CHR$(C1);"                     "
250  PRINT TAB(K);CHR$(C1);"                     "
270  PRINT :K=16
280  PRINT TAB(K);CHR$(C6);"                 "
290  PRINT TAB(K);CHR$(C6);"                 "
300  PRINT TAB(K);CHR$(C6);"                 "
310  PRINT TAB(K);CHR$(C6);"                 "
320  PRINT TAB(K);CHR$(C6);"                 "
340  PRINT :PRINT :K=13
350  PRINT TAB(K);CHR$(C3);"                     "
360  PRINT TAB(K);CHR$(C3);"                     "
370  PRINT TAB(K);CHR$(C3);"                     "
380  PRINT TAB(K);CHR$(C3);"                     "
390  PRINT TAB(K);CHR$(C3);"                     "
```



```
410  LINE (83,6)-(527,193),C5-248,B
420  LINE (75,3)-(535,196),C5-248,B
430  LINE (70,1)-(540,198),C5-248,B
460  YHEL=4 :NBMB=4 :XWLL=27 :YWLL=22 :FALSE=0 :TRUE=NOT FALSE :GOSUB 5000
470  FOR I=1 TO 1500:NEXT I:CLS:GOSUB 1570:LOCATE 10,5:PRINT CHR$(C2);"ﾄﾞｸﾛ=ﾄｳ"
510   LOCATE 50,5 :PRINT CHR$(C4);"ﾍﾘｺﾌﾟﾀｰ"
520   LOCATE 50,7 :PRINT CHR$(C4);" ──┬── "
530   LOCATE 50,8 :RHEL$="  ◥■■● " :PRINT CHR$(C4);RHEL$
550 '
560 ''560->600 ｻｸｼﾞｮ
630   LOCATE 53,10 :PRINT CHR$(C2);"●";"   ﾊﾞｸﾀﾞﾝ"
650 '
700   LOCATE 60,19 :PRINT CHR$(C2);"ｾﾝｽｲｶﾝ";CHR$(C1);
760   LOCATE XWLL+2,YWLL :PRINT CHR$(C6);"ﾎﾞｳｷﾞｮｾﾝ";CHR$(C1);
840   LOCATE 22,19 :PRINT CHR$(C2);"ｱｼ";CHR$(C1);
900   LOCATE 50,7 :PRINT CHR$(C4);" ──┬── "
910   LOCATE 50,8 :LHEL$=" ●■■◤  " :PRINT  CHR$(C4);LHEL$
960  CLS :GOSUB 3300
1035  FOR I=0 TO 2 :PNT(I)=0 :FOR J=0 TO 7 :FOOT$(J,I)="■" :NEXT J: NEXT I
1040  COLOR ,0 :CLS :GOSUB 1570 :GOSUB 1800 :' ﾄﾞｸﾛ=ﾄｳ ｦ ｶｸ, ｾﾝｽｲｶﾝ ｦ ｶｸ
1070  LINE (0,20)-(639,20),C7-248
1080  LOCATE 59,1 :PRINT CHR$(C7);"TIME";TM
1100  LOCATE 40,10 :PRINT CHR$(C6);"START"
1110  FOR I=1 TO 5 :SOUND 69,18 :FOR J=1 TO 50 :NEXT J :NEXT I
1120  LOCATE 40,10 :PRINT CHR$(C0);"START"
1160   IF I$=" " THEN GOSUB 1870 :IF XBEEM=48 THEN GOSUB 2830 ELSE 1220
1300  COLOR C6-248,C2-248 :CLS :K=12 :LOCATE 0,4
1500  COLOR C7-248,0 :CLS
1520   SOUND 68,18 :PRINT :PRINT :PRINT :PRINT "     ｻｲﾄﾞ ﾄﾗｲ ｽﾙｶ (Y/N) ?";
1640  FOR I=9 TO 0 STEP -1
1650   LOCATE 0,7+I
1655   PRINT CHR$(C2);
1660   FOR J=0 TO 3
1670    PRINT DK$(I,J);
1680   NEXT J
1685   PRINT CHR$(C0);
1690  NEXT I
1700 ''1700->1740, 3220 ｻｸｼﾞｮ
1760  FOR I=24 TO 17 STEP -1 :LOCATE XWLL,I :PRINT CHR$(C6);"■";CHR$(C1); :NEXT
I
1810  COLOR C1-248
1815  LOCATE XOLDSBM,YOLDSBM   :PRINT "■■■■■■";
1816  LOCATE XOLDSBM,YOLDSBM+1 :PRINT "■■■■■■";
1830  LOCATE XSBM,YSBM   :PRINT "■■■◤";CHR$(C1);
1835  LOCATE XSBM,YSBM+1 :PRINT "◤      ";
1880  XBEEM=(XWLL+1)*8
1920   COLOR C6-248 :GOSUB 8000
1930   SOUND 68,18
1940   COLOR C1-248 :GOSUB 8500
1980    COLOR C6-248 :GOSUB 8000
1990    SOUND 68,18
2000    COLOR C6-248 :GOSUB 8500
2010    FOR I=21 TO XBEEM/8 STEP -5
2020     LOCATE I,YBEEM/8 :PRINT " ";
2025     IF I>9 THEN FOOT$( (PNT(I/5-2)+YSBM-17) MOD 8,I/5-2 )=" "
2090   LOCATE XSBM,YSBM   :PRINT CHR$(C1);"■■■■■■";
2095   LOCATE XSBM,YSBM+1 :PRINT CHR$(C1);"■■■■■■";
2153  LOCATE XWLL,YWLL-1 :PRINT CHR$(C6);"■";
2154  LOCATE XWLL,YWLL   :PRINT CHR$(C6);" ";
2155  LOCATE XWLL,YWLL+1 :PRINT CHR$(C6);" ";
2160  LOCATE XWLL,YWLL+2 :PRINT CHR$(C6);"■";
2200  FTILND=INT(RND(1)*3)
2205 '
2210  PNT(FTILND)=PNT(FTILND)-1 :IF PNT(FTILND)<0 THEN PNT(FTILND)=7
2216  FOOT$( PNT(FTILND),FTILND )="■"
2220  FOR K=17 TO 24
2225   LOCATE FTILND*5+10,K :PRINT CHR$(C2);FOOT$( PNT(FTILND),FTILND );
2227   PNT(FTILND)=( PNT(FTILND)+1 ) MOD 8
2228  NEXT K
2280   LOCATE XHEL,YHEL :PRINT CHR$(C4);" ──┬── "
2290   LOCATE XHEL,YHEL+1:IF DR=1 THEN PRINT CHR$(C4);RHEL$ ELSE PRINT CHR$(C4);
LHEL$
2440    IF YB(I)>16 THEN 2520 ELSE CC=C0
2450    LOCATE XB(I),YB(I) :PRINT CHR$(CC);"■";
2460    YB(I)=YB(I)+1 :IF YB(I)>=16 THEN CC=C1
2480     LOCATE XB(I),YB(I) :PRINT CHR$(C2);"●";CHR$(CC);
2530  CC=C1
2540  IF NOT ( CHR$( SCREEN( XB(I),YB(I)+1 ) )=" "  OR  CHR$( SCREEN( XB(I),YB(I
)+1 ) )="◤" )  THEN 2450
2600    SOUND 68,18
2610    FOR J=1 TO 25 :NEXT J
2660    GOSUB 2790 :LINE (X1,Y1)-(X,Y),C2-248
2700    CC=C2-248 :IF K=2 THEN CC=C6-248 :R=37
2720     GOSUB 2790 :PSET (X,Y),CC
2770  RETURN 1290
2840  COLOR C2-248
2850  XILND=21 :YILND=9 :DKR=3 :GOSUB 6500 :GOSUB 6000 :'RIGHT 5 BLOCKS
2860  '
2870  YILND=11 :GOSUB 6500 :GOSUB 6000 :'ALSO,RIGHT 5 BLOCKS
2880  XILND=16 :YILND=9 :DKR=2 :GOSUB 6500 :GOSUB 6000 :' NEXT RIGHT 5 BLOCKS
2890  '
2900  XILND=21 :YILND=13 :DKR=3 :GOSUB 6500 :GOSUB 6000 :' RIGHT 5 BLOCKS
2910  XILND=16 :YILND=11 :DKR=2 :GOSUB 6500 :GOSUB 6000 :' NEXT RIGHT 5 BLOCKS
2920  XILND=11 :YILND=9 :DKR=1 :GOSUB 6500 :GOSUB 6000 :' NEXT LEFT 5 BLOCKS
2930  '
2940  FOR I=8 TO 24 STEP 2
2950   XILND=21 :YILND=I+6 :DKR=3 :GOSUB 6500 :GOSUB 6000
2960   XILND=16 :YILND=I+4 :DKR=2 :GOSUB 6500 :GOSUB 6000
2970   XILND=11 :YILND=I+2 :DKR=1 :GOSUB 6500 :GOSUB 6000
2980   XILND=1 :YILND=I   :DKR=0 :GOSUB 6000
2985   XILND=8 :GOSUB 6500: PRINT "   ";
2990  NEXT I
2995  LOCATE 1,24 :PRINT SPACE$(26);
3060  RETURN 980
3130   SOUND 68,18
3140   COLOR C6-248 :LOCATE 35,4 :PRINT "ｼﾞｶﾝｷﾞﾚ !!!"
3150   FOR K=1 TO 10 :SOUND 68,18 :FOR J=1 TO 50 :NEXT J :NEXT K
3170   RETURN 1290
3200  SOUND 68,18
3260  FOR I=0 TO 30 :I$=INKEY$ :NEXT I
3310  SOUND 68,18 :LOCATE 20,8 :PRINT CHR$(C7);"ｱｿﾋﾞｶﾀ ﾉ ｾﾂﾒｲ ｶﾞ ﾋﾂﾖｳ ﾃﾞｽｶ (Y/N)
 ?";
3320  GOSUB 3250 :ANS$=I$ :PRINT ANS$
3330  IF ( ANS$<>"Y" ) AND ( ANS$<>"y" ) THEN RETURN
3340   CLS :LOCATE 24,2 :PRINT CHR$(C5);"***"
3350   LOCATE 45,2 :PRINT "***"
3360   LOCATE 29,2 :PRINT CHR$(C6);"ｱｿﾋﾞｶﾀ ﾉ ｾﾂﾒｲ";CHR$(C5);
3370 '
3380   LOCATE 13,6 :PRINT CHR$(C7);"ｺﾉ ｹﾞｰﾑ ﾊ, ｷﾒﾗﾚﾀ ｼﾞｶﾝ ﾅｲ ﾆ";
3390   PRINT CHR$(C2);"'ﾄﾞｸﾛ=ｷﾁ'";
3400   PRINT CHR$(C7);"ｦ ﾊｶｲ ｽﾙ ｹﾞｰﾑ ﾃﾞｽ｡"
3410   GOSUB 3940
3420 '
3430   LOCATE 6,9 :PRINT CHR$(C7);"ｱﾅﾀ ﾊ,"; :PRINT CHR$(C5);"'ｸﾛ ﾉ ｾﾝｽｲｶﾝ'";
3440   PRINT CHR$(C7);"ｦ ﾂｶｲ,"; :PRINT CHR$(C6);"'ﾋﾞｰﾑ ﾎｳ'";
3450   PRINT CHR$(C7);"ﾃﾞ"; :PRINT CHR$(C2);"'ﾄﾞｸﾛ=ﾄｳ' ﾉ ｱｼ";
3460   PRINT CHR$(C7);"ｦ ｺｳｹﾞｷ ｼﾏｽ｡"
3470   GOSUB 3940
3480 '
3490   LOCATE 6,11 :PRINT CHR$(C2);"'ﾄﾞｸﾛ=ﾄｳ' ﾉ ｱｼ";
3500   PRINT CHR$(C7);"ｦ, 4 ﾎﾝ ﾄﾓ";
3510   PRINT CHR$(C6);"'ﾋﾞｰﾑ ﾎｳ'"; :PRINT CHR$(C7);"ﾃﾞ ﾊｶｲｽﾙﾄ,";
3520   PRINT CHR$(C3);"ｱﾅﾀ ﾉ ｶﾁ"; :PRINT CHR$(C7);"ﾃﾞｽ｡"
3530   GOSUB 3940
3540 '
3550   LOCATE 6,14 :PRINT CHR$(C7);"ﾃｷ ﾊ,"; :PRINT CHR$(C4);"'ﾍﾘｺﾌﾟﾀｰ'";
3560   PRINT CHR$(C7);"ｶﾗ";
3570   PRINT CHR$(C6);"'ﾊﾞｸﾀﾞﾝ'"; :PRINT CHR$(C7);"ｦ ｵﾄｼﾃ, ｺｳｹﾞｷ ｼﾃ ｷﾏｽ｡"
3580   GOSUB 3940
3590 '
3600   LOCATE 12,17 :PRINT CHR$(C7);"ｳﾏｸ ｶﾜｼﾅｶﾞﾗ, ﾃｷ ｦ ｺｳｹﾞｷ ｼﾃ ｸﾀﾞｻｲ｡"
3610   GOSUB 3940
3620 '
3630   LOCATE 20,21 :PRINT CHR$(C3);"ﾂｷﾞﾆ, Key ﾉ ｾﾂﾒｲ ｦ ｼﾏｽ｡"
3640   LOCATE 65,23 :PRINT CHR$(C2);"Hit Any Key !" :GOSUB 3250
3650 '
3660  REM   Key ﾉ ｾﾂﾒｲ
3670   CLS :LOCATE 24,2 :PRINT CHR$(C4);"***"
3680   LOCATE 42,2 :PRINT "***"
3690   LOCATE 29,2 :PRINT CHR$(C6);"Key ﾉ ｾﾂﾒｲ";CHR$(C4);
3700 '
3710   LOCATE 17,8 :PRINT CHR$(C6);"'ﾋﾞｰﾑ ﾎｳ'"
3720   LOCATE 50,7 :PRINT CHR$(C5);"'ｾﾝｽｲｶﾝ'"; :PRINT CHR$(C7);"ｦ ｳｺﾞｶｽ"
3730 '
3740   LOCATE 14,11 :PRINT CHR$(C6);"■■■■■■■■■■"
3750   LOCATE 14,13 :PRINT CHR$(C6);"( ｽﾍﾟｰｽ ﾊﾞｰ )"
3760 '
3770   LOCATE 44,10 :PRINT CHR$(C3);"           ( ｳｴ )"
3780   LOCATE 44,12 :PRINT CHR$(C7);"              8"
3790   LOCATE 44,14 :PRINT CHR$(C2);"              I"
3800   LOCATE 44,15 :PRINT CHR$(C2);"              I"
3810 '
3820   LOCATE 42,16 :PRINT CHR$(C3);"( ﾋﾀﾞﾘ )"; :PRINT CHR$(C7);" 4 ";
3830   PRINT CHR$(C2);"---+---"; :PRINT CHR$(C7);" 6 ";
3840   PRINT CHR$(C3);"( ﾐｷﾞ )"
3850 '
3860   LOCATE 44,17 :PRINT CHR$(C2);"              I"
3870   LOCATE 44,18 :PRINT CHR$(C2);"              I"
3880   LOCATE 44,20 :PRINT CHR$(C7);"              2"
3890   LOCATE 44,22 :PRINT CHR$(C3);"            ( ｼﾀ )"
3900 '
3910   LOCATE 59,24 :PRINT CHR$(C4);"Hit Any Key !"; :GOSUB 3250
3950  FOR I= 1 TO 3000
4000  DATA  248, 249, 250, 251, 252, 253, 254, 255
5000 '..... 5000->7100 ﾊ, Level III ﾄ ｵﾅｼﾞ Subroutine ﾃﾞｽ .....
8000 '
8010 REM ----- ﾋﾞｰﾑ 2 -----
8020  FOR I=XSBM+3 TO XWLL+1 SETP -1
8030   LOCATE I,YSBM :PRINT "─";
8040  NEXT I
8050  RETURN
8500 '
8510 REM ----- ﾋﾞｰﾑ ｦ ｹｽ -----
8520  FOR I=XSBM+3 TO XBEEM/8+1 STEP -1
8530   LOCATE I,YSBM :PRINT "■";
8540  NEXT I
8550  LOCATE XWLL+2,YSBM :PRINT CHR$(C1);
8560  FOR J=22 TO 7 STEP -5
8570   LOCATE J,YSBM :PRINT CHR$(C1);
8580  NEXT J
8590  RETURN
```

上記の行だけ変更してください。

マイコンとキミを結ぶビジュアル・インターフェース―マイコン入門誌

MICOMBOY

次号予告

マイコンボーイ

Ａ４判 160頁　予価 530円

■夏休み前のビッグ・プレゼント――― 第2号は，7月8日発売予定！

マイコン雑誌は，無味乾燥なものが多いといわれます。また，初級入門者を切りすてているような雑誌も多いといわれています。一誌一機種という雑誌も多くあります。

しかし，本誌のように多機種にコンバージョンされ活用できるユニークな雑誌は，ほとんどありません。

次号も，FUJITSU, TOSHIBA, NEC, HITACHI, SHARP などの機種を予定しています。

ストーリーを追ってプログラミングができるユニークなマイコン入門誌≪マイコンボーイ≫は，第２号にむけて速力全開でスタートしています。

本号で大活躍した≪乱≫，≪エルシー≫≪ランダム≫，≪タブ≫らも，新しい強敵に向って猛勉強を開始しました。

さらに，第２号では，魅力あふれるキャラクターや，笑いころげさせるギャグマンが登場してきます。

≪主な内容≫

- ▶マイコンのさわり方を解説した≪イントロダクションのページ≫
- ▶アクションマンガと対応させた≪シミュレーションゲームのページ≫
- ▶おもしろさの中で学んでいける≪ギャグマンガゲームのページ≫
- ▶マイコンに関する質問や，ゆかいなパロディ風の話などを中心とした≪読者と編集部をつなぐページ≫

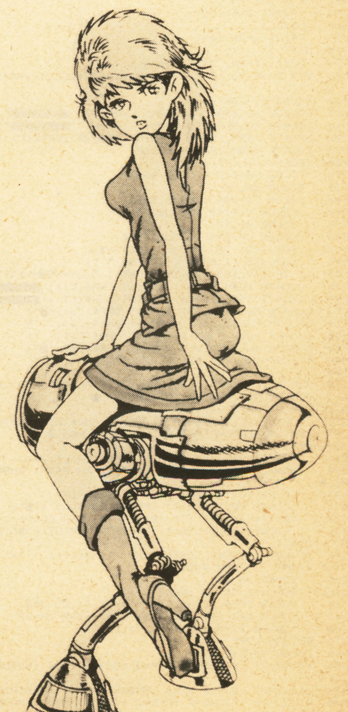
投稿大歓迎！

プログラム＆イラスト

マイコンボーイ編集部では，ご意見イラスト，プログラムのご投稿をお待ちしています。ご意見やイラストは官製ハガキで，プログラムは，カセットテープおよびプログラムの概略説明書付でご投稿ください。ご意見，イラストで本誌に掲載された方には，〈記念品〉を贈呈します。また，プログラムは，編集部で審査したうえ，ストーリー化できるものは，マンガ家とタイアップして誌上に掲載し，規定の稿料をお支払い致します。また，それ以外でもすぐれたソフトは，販売契約を結ぶことがありますので，ふるってご投稿ください。ただし，他誌に未掲載のオリジナルなものに限ります。







ベーシック・アドベンチャー。次回はいよいよ シミュレーション・ワールドへ　ワープ！

Obunsha

マイコンボーイ アンケート

● この本の印象について

	難しさ	興味度	満足度
Ⅰ この本の全体について			
Ⅱ 電子地雷エレクトロマイン			
Ⅲ スロットボンバー			
Ⅳ レザー・サブマリン			

● この本についてのご意見をお聞かせください。

● 今後，マイコン関係でどんな内容の本をお望みですか？

①BASICの入門書　②ゲームのプログラム
③マイコン全体についての入門書　④マイコンについての最新情報
⑤マイコンのソフト集

⑥その他

あなたがお持ちの周辺機器	①フロッピー ②プリンター ③テープレコーダー ④カラーCRT ⑤グリーンCRT	主に使用されているプログラム言語	①BASIC　②PASCAL ③FORTRAN　④COBOL ⑤その他 〔　　　〕
この本をどこで知りましたか	①中一時代　②中二時代　③中三時代　④高一時代　⑤高二時代　⑥螢雪時代　⑦夕刊フジ　⑧日刊ゲンダイ　⑨朝日新聞　⑩毎日新聞　⑪読売新聞　⑫日本経済新聞　⑬日経産業新聞　⑭毎日中学生新聞　⑮マイコン BASIC MAGAZIEN　⑯友だちに聞いた　⑰書店のチラシで	⑱その他〔　　　〕	

アンケートの記入方法

左のⅠ～Ⅳの項目について，それぞれ，下の5段階の評価をし，A～Eの記号でお答えください。

1. 難しさ
   - A：大変むずかしかった
   - B：ややむずかしかった
   - C：ふつう
   - D：やさしすぎた
   - E：まったく易しすぎて不満
2. 興味度
   - A：大変おもしろかった
   - B：まあ，おもしろかった
   - C：ふつう
   - D：あまり，おもしろくなかった
   - E：ぜんぜんおもしろくなかった
3. 満足度
   - A：大変満足した
   - B：まあ，満足した
   - C：ふつう
   - D：やや不満
   - E：まったく不満

このアンケートにお答えいただいた方に
すばらしいカメラが当たります!!

※読者のみなさんのご意見を今後の編集の参考にさせていただきたいと思います。ぜひご協力ください。ご協力いただいた方の中から抽選により3名の方に、ストロボ付、オートワインダー式ポケットカメラをさしあげます。アンケートの内容によって抽選が左右されることはありません。正確にご記入くださいますようお願いいたします。

(発表は賞品の発送をもってかえさせていただきます。)

(キリトリ線)

郵便はがき

料金受取人払

162-□□

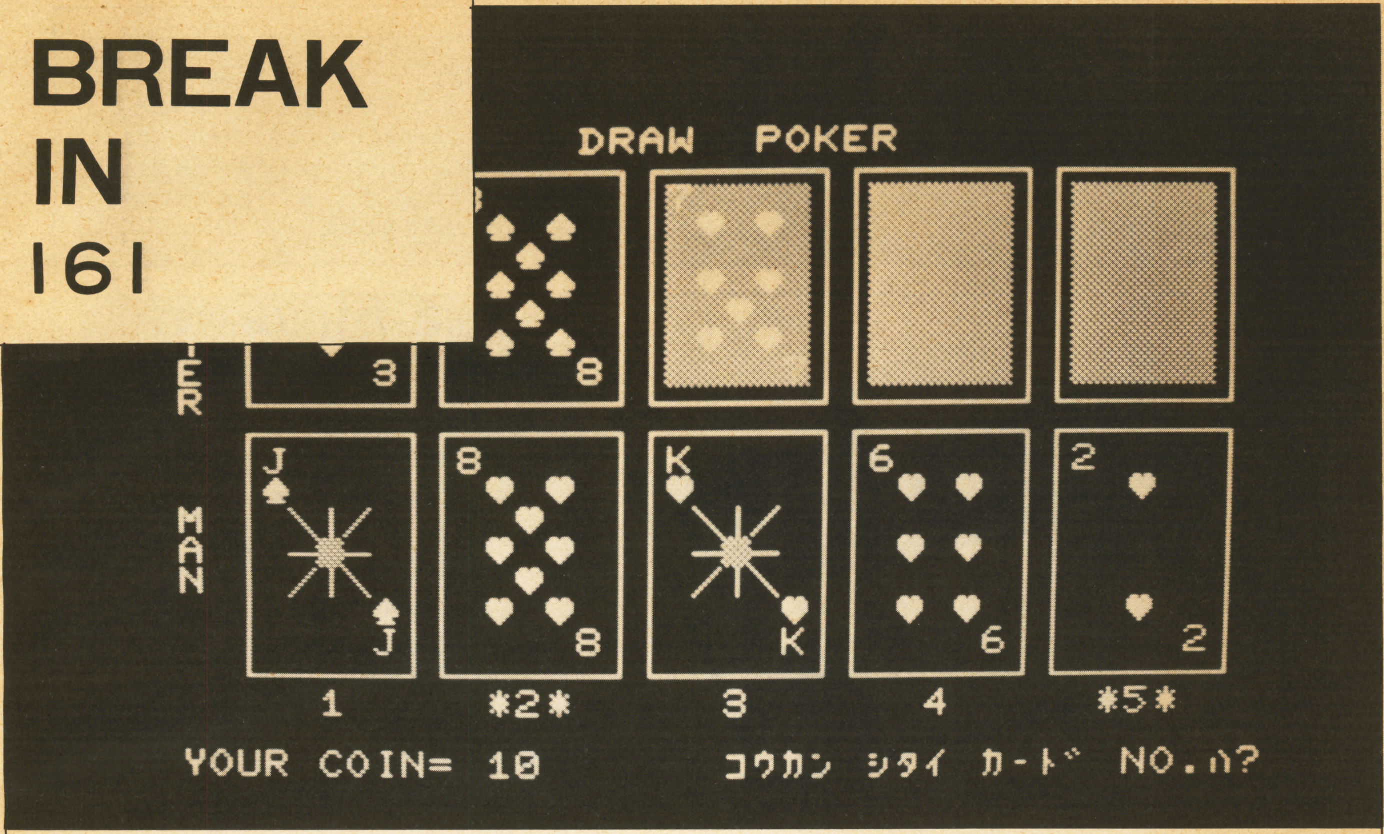
牛込局承認
361

差出有効期間
昭和58年11月
30日まで
郵便切手はいりません

東京都新宿区横寺町55
旺文社 雑誌局
マイコンボーイ編集部

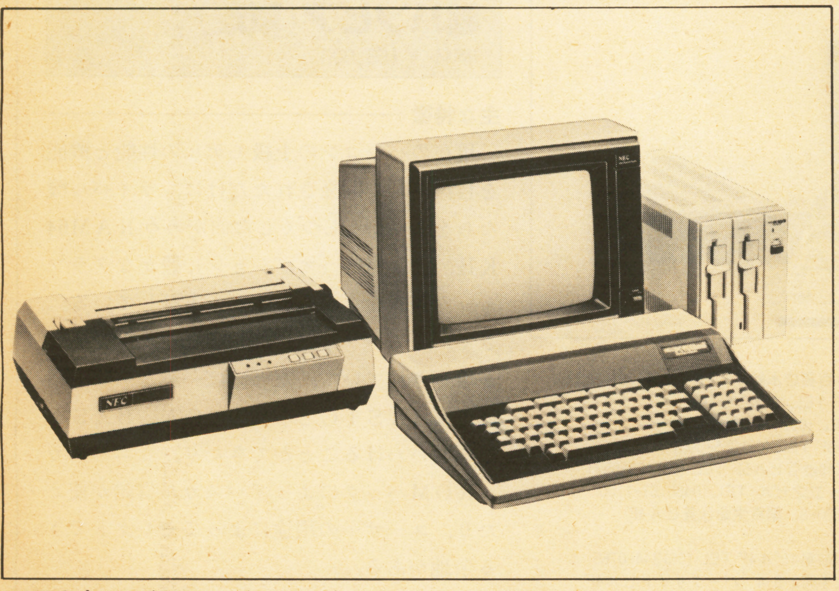
マイコンをお持ちですか	①はい ②いいえ	機種名	①PC−6000 ②PC−8000 ③PC−8800 ④FM−7 ⑤FM−8 ⑥FM−11 ⑦アップルII ⑧パソピア ⑨MZ700 ⑩日立レベル3 ⑪その他 ( )
いいえとお答えの方。購入予定は?	①あり ②なし ↓ 機種名 ( )		
主にどんな用途に使用していますか?	①ゲーム ②仕事 仕事の内容 ( )	ご家族であなたの他にマイコンに興味をお持ちの方	①父 ②母 ③兄 ④弟 ⑤息子 ⑥娘 ⑦その他
性別 年齢	①男 ②女 ( )歳	勤務先・学校名 所属部課・学科名 電話	☎( ) −
ご住所	〒 ☎( ) −		
ご氏名			

該当する番号に○を付けてください。この欄は保存用にコピーして直接使用されます。大きく,わかりやすく,ていねいにお願いいたします。 ⇨



マイコン誌上カタログ セレクト11

最新のニュー・モデルを中心に，ゲームから実務まで活用できるマイコンを紹介。



●NECパソコンインフォメーションセンター
〒108 東京都港区三田3-14-10 明治生命三田ビル TEL (03) 452-8000(代)
●NECショールームC&Cプラザ
〒100 東京都千代田区内幸町2-2-3 日比谷国際ビル TEL (03) 595-0511(代)
●BIT-INNシステムセンター
東京＝〒101 東京都千代田区外神田1-15-16 ラジオ会館7F TEL (03) 255-4006，4575〜6
横浜＝〒220 横浜市西区北幸1-8-4 横浜西口第2ミナトビル7F TEL (045) 314-7707〜9
名古屋＝〒460 名古屋市中区大須3-31-22 明治生命上前津ビル1F・2F TEL (052) 263-0971
大阪＝〒542 大阪市南区久佐衛門町24 難波日興ビル9F TEL (06) 213-8864
●NECマイコンショップ
全国に約200ヵ所ありますのでお気軽に。

■PC-8001mkⅡ(NEC)

本体￥123,000

主な特徴

名機，またはベストセラー機であったＰＣ－8001のグレードアップ・モデルだ。価格は大幅にダウンしているけれど，内容は大幅にアップだ。ＰＣ－8001の実力は知っていても，いまや旧式であるから二の足を踏んでいた人にとって，期待の「マークⅡ」かも知れない。まず，64ＫＢのＲＡＭを実装(従来は16ＫＢ)するとともに，インターフェースの強化をはかり，漢字対応（オプション）も実現。とにかく，価格を下げて，内容は逆に充実という，「マークⅡ」の公式をしっかりとはたしている。もちろん従来機のソフトプログラムをそっくり受け継げるメリットは，このメカの価値を何倍にも高めている。

●ＣＰＵ：μＰＤ 780 Ｃ-1（Ｚ－80Ａコンパチブル）●ＲＯＭ：32ＫＢ●ＲＡＭ：64ＫＢ(ＢＡＳＩＣ時は32ＫＢ，グラフィックメモリー16ＫＢ)

マイコン ミニ用語

●**CPU**（シーピーユー）
Central Processig Unit の略で「中央処理装置」と訳され，演算や制御などコンピュータの中で中心的な役割をする。情報を出し入れする入出力装置，情報をしまっておくメモリと一体となって，マイコンとして動作する。

●**RAM**（ラム）
使う人（ユーザー）が自由に使えるメモリのことで Random access memory を略してRAMと呼んでいる。ここにはプログラムやデータを記憶させるため，この容量が多ければ多いほど，多くの仕事や複雑な仕事をさせることができるわけだ。



●シャープOAショールーム
札幌＝〒063 札幌市西区二十四軒1条7-3-17　シャープ札幌ビル
（58年５月開場予定）
仙台＝〒983 仙台市六丁目字本新田東1　シャープ仙台ビル TEL (0222) 88-9151
東京・市ケ谷＝〒162 新宿区市ケ谷八幡町8　シャープ東京ビル TEL (03) 260-1161（代）
横浜＝〒231 横浜市中区本町4-43　横浜三菱商事ビル TEL (045) 201-4371
名古屋＝〒454 名古屋市中川区山王3-5-5　シャープ名古屋ビル TEL (052) 332-2631（代）
大阪＝〒545 大阪市阿倍野区長池町22番22号　シャープ本社ビル TEL (06) 621-1221（代）
広島＝広島市中区八丁堀1-8　8（エイト）ビル
（58年５月開場予定）
福岡＝〒812 福岡市博多区博多駅前1-98　博多駅前第1生命ビル
（58年７月開場予定）

■MZ-731（シャープ）
本体 ￥128,000

主な特徴

MZ-700シリーズのトップモデルで，プリンター，カセットレコーダーが一体化されたオールインワン・タイプ。最新モデルだけあって内容的にひじょうに充実していて，１～２年前の15～20万円クラスと同等の性能と機能をもっているといっていい。メインメモリは64KBと大きく，ゲーム，ホビーだけでなく実務にも十分な能力をもつ。このパソコンは，シャープのパソコンに共通するクリーン思想を受け継ぎ，システムチェンジを可能にしている。

内蔵のプリンターは，いわゆるカラープロッタプリンターで，グラフや各種図形のプリント・アウトを可能にしている。

●CPU：Z80A●ROM：４KB（モニタ），４KB（キャラクタジェネレータ）●RAM：64KB（VRAM：４KB）



●東芝パソコンサロン
札幌＝㈱ディテール：〒064 札幌市中央区南一条西２丁目　丸井今井百貨店本館４F TEL (011) 221-5023
仙台＝丸菱倉庫㈱：〒980 仙台市木町通2-1-55　第八丸昌興業ビル2F TEL (0222) 75-2203
東京・銀座＝緑屋電気㈱：〒104 中央区銀座5-9-1　銀座幸ビル6F TEL (03) 574-0941
東京・秋葉原＝田中電気㈱：〒101 千代田区外神田1-16-9　朝風ビル8F TEL (03) 255-0901
東京・渋谷＝日本情報通信システム㈱：〒150 渋谷区渋谷1-12-7　三和渋谷ビル5F TEL (03) 499-5571
埼玉・大宮＝立之システムリサーチ㈱：〒330 大宮市櫛引町2-285-1　立之ビル TEL (0486) 51-1100
名古屋＝日本ビジネスオートメーション㈱：〒460 名古屋市中区錦1-20-8　第百生命伏見ビル4F
TEL (052) 202-1048
大阪＝大阪パソピア㈱：〒530 大阪市北区梅田1-11-4-200号　大阪駅前第4ビル4F TEL (06) 344-0765
広島＝東芝ビジネスコンピュータ㈱広島営業所：〒730 広島市中区大手町2-8-4　パークサイドビル4F
TEL (082) 249-6762
福岡＝㈱九州電子機器サービス：〒810 福岡市中央区大名2-7-11　斉藤ビル3F TEL (092) 711-1915
●東芝OAショールーム
札幌＝〒060 札幌市中央区北三条西1丁目　東芝札幌ビル1F TEL (011) 214-2545
東京・虎ノ門＝〒105 港区虎ノ門1-21-17　虎の門NNビル1F TEL (03) 508-9375（代）
東京・新宿＝〒160 新宿区西新宿2-4-1　新宿NSビル5F・OAセンター TEL (03) 342-3295
長野・松本＝〒390 松本市中央2-6-1　住友生命松本ビル TEL (0263) 35-6610
名古屋＝〒460 名古屋市中区栄2-10-19　商工会議所ビル8F TEL (052) 202-8730（代）
大阪＝〒550 大阪市西区靭本町1-11-7　信濃橋三井ビル1F TEL (06) 445-4980
広島＝〒730 広島市中区大手町2-8-4　パークサイドビル4F TEL (082) 249-4050

■パソピア（東芝）
本体 ￥163,000

主な特徴

現在のパソコンの上級モデルの性能と機能を目標として開発されただけに，ライバル機たちをよく研究しつくした面がある。RAM64KB，漢字対応（オプション），豊かなグラフィック機能などがそのことを実証している。さて，このパソピアには，多くのオプションが用意されていて，たとえば一般ユーザー用のT－BASICと事務用のOA－BASICを使いわけることができたり，ミニパスカルROMパックによってパスカル言語が使えたり，ユーザーの用途によってセレクトできる余地がある。

●CPU：Z－80A●ROM：32KB●RAM：64KB

マイコンミニ用語

●ROM（ロム）

Read Only Memoryの略。ラムと異なり，電源を切ってもメモリが消えないし，メモリの内容を変更することもできない「読み出し専用」メモリのこと。ラムが内容変更できる音楽テープなら，ロムは変更できないレコードといったところ。

●コンパチブル

互換性があるということ。現在，他社間はもちろん同社間でもソフトの互換性をもつ機種は少ないが，改良モデルや姉妹モデル間では互換性をもたせる傾向になってきている。マークⅡは，従来モデルとかなりソフトコンパチだ，などと表現される。



■ベーシックマスターL３/マークⅡ(日立)

本体 ¥198,000

主な特徴

従来機のＬ３のニュー・バージョン。機能や性能を向上させながら価格を引き下げている。主な改良点としては，別売オプションによる漢字対応をはかったことだが，トータルな面で改善しているため，より一層の汎用性を実現している。

このＬ３／マークⅡは，もちろんホビーにも使えるけれど，形状その他の面でむしろ実務分野で評価が高い。

本機に限らず「マークⅡ」のメリットは，豊富なソフトプログラムが使える点で，パソコンを手にしたときから，先代の財産を受け継げる点だ。

●ＣＰＵ：6809●ＲＯＭ：24ＫＢ（モニタおよびＢＡＳＩＣ）●ＲＡＭ：32ＫＢ

●日立パソコンショールーム

東京＝〒104 東京都中央区銀座西2-2　有楽フードセンター東館1F（ローディプラザ内） TEL (03) 562-1340
名古屋＝〒460 名古屋市中区栄4-1-7　朝日生命会館5F TEL (052) 251-4111
広島＝〒730 広島市中区基町11-13　第一生命ビル TEL (082) 223-4111
福岡＝〒812 福岡市博多区店屋町7-18　博多渡辺ビル TEL (092) 291-2323

●日立ＯＡショールーム

仙台＝〒980 仙台市一番町3-6-1　佐々重ビル TEL (0222) 24-3062
東京・新宿＝〒160 東京都新宿区西新橋2-4-1　新宿NSビル TEL (03) 344-4671
東京・鎌倉橋＝〒101 東京都千代田区内神田1-1-14　日立鎌倉橋別館 TEL (03) 295-2045
富山＝〒930 富山市総曲輪2-1-3　富山商工会議所ビル TEL (0764) 25-7369
名古屋＝〒460 名古屋市中区新栄町2-14 TEL (052) 962-2414
大阪＝〒530 大阪市北区堂山町3-3　日生梅田ビル TEL (06) 315-1956
高松＝〒760 高松市天神町10-12　栄泉高松ビル TEL (0878) 31-2111
（注：広島・福岡はパソコンショールームと同じ住所）



■FM-7(富士通)

本体 ¥126,000

主な特徴

大きなメモリーと豊かなグラフィック機能を特徴としたＦＭ－８のバリエーションモデルだ。価格の面からはＦＭ－７が弟機のように思われがちだけど，内容的にはほとんど差がなく，むしろ実用性を向上させ，使いやすく，買いやすくなったことに注目したい。

この価格でメモリーが64ＫＢで，漢字対応（オプション），640×200ドット（８色）は，このクラスでは抜群といっていい。

ＦＭ－８のソフトプログラムは，質と量の点で評判だけど，ＦＭ－７はこれとコンパチブルになっている。

●ＣＰＵ：ＭＢＬ68Ｂ09，Ｚ80Ａ（オプション）●ＲＯＭ：32ＫＢ●ＲＡＭ：64ＫＢ（ＢＡＳＩＣ時：32ＫＢ＋32ＫＢ，ＶＲＡＭ：48ＫＢ）

●富士通マイコンスカイラブ

札幌＝〒060 札幌市中央区南1条西2-11　丸井今井1条本館4F TEL (011)241-4185
東京・虎の門＝〒105 港区虎ノ門2-3-13　第18森ビル8F TEL (03)591-1091/2561
東京・秋葉原＝〒101 千代田区外神田1-15-16　秋葉原ラジオ会館6F TEL (03) 251-1448/1449
名古屋＝〒460 名古屋市中区栄1-5-22 TEL (052) 221-6016
大阪＝〒530 大阪市北区梅田1-2-2　大阪駅前第2ビル1F TEL (06) 344-7628
広島＝〒730 広島市中区立町4-2　大橋ビル2F・3F TEL (082) 247-3949

マイコンミニ用語

●ビット
ビットとは，ONかOFFか,あるかないか, YESかNOか…。これが情報の最低の単位だ。bitはbinary（2進の）digit（桁）の略で，8ビットといえば，2進数の8桁で表される情報の大きさだ。

●バイト
1ビットでは2通りしか表せないので，数ビットをまとめて数多くの状態を表すと，複雑な表現ができる。ビットの数を表現する場合の単位としてバイトがあり，8ビットを1バイトという。1バイトは2^8＝256通りの状態が表現できる。



FP-1100（カシオ）
本体 ￥128,000

主な特徴

CPUのデュアル構成による高速処理と大容量メモリー（ユーザーズエリアで64KB）などは，このクラスの水準を超える。従来，カシオのマイコンは，ビジネス機としてのイメージが強く，一般のユーザーにはなじまなかったけれど，本機はグラフィック機能の強化とあいまって，ホビーからビジネスのすべてをこなす，本来のパソコンとなっている。
●CPU：Z80A●ROM：36KB（サブ：8KB）●RAM：64KB（サブ：48KB）



コモドール64（コモドール）
￥99,800

主な特徴

コモドールには名器VIC-1001があるが，本機はその基本コンセプトを受け継ぎながらも，内容的には大幅な強化をはかり，本格的なホームコンピュータとして登場させた。とにかく64KB RAMを実装していることからもわかるように，VICのイメージを一掃する高性能・多機能マシーンだ。しかし，ゲームカートリッジを用意するなど，あいかわらずゲーム派にもキメ細かな気配りをしている。
●CPU：6510●ROM：20KB●RAM：64KB



M5（ソード）
本体￥49,800

主な特徴

学習用からゲーム用途を目的にしたB5サイズのホーム・コンピュータ。ユニークなのは、ベーシックが3種類用意されていることで，用途に応じて使いわける。その中のひとつであるBASIC−Gでは，16色で絵が描け，アニメーションも楽しめる。とにかく気軽にパソコンに接して，ゲームや音楽を楽しむのに最適だ。
●CPU：Z80A●ROM：8KB（内蔵モニタ）
●RAM：20KB（内VRAM：16KB）

マイコンミニ用語

●周辺機器
一般にキーボードとＣＰＵ（入出力装置やメモリも含む）を本体といい，外部につけるプリンター，補助記憶装置（フロッピーディスクユニット，データレコーダー），ディスプレイなどを周辺機器という。周辺機器の方が高価になる場合が多い。

●拡張
一般に機能の向上，能力の増大をはかるときに使われる用語。たとえば，増設ＲＡＭを付加して，メモリ16ＫＢを32ＫＢにユーザーエリアを拡大することも拡張だし，音楽機能のない機種にサウンド（音楽）機能を加えることも拡張などという。

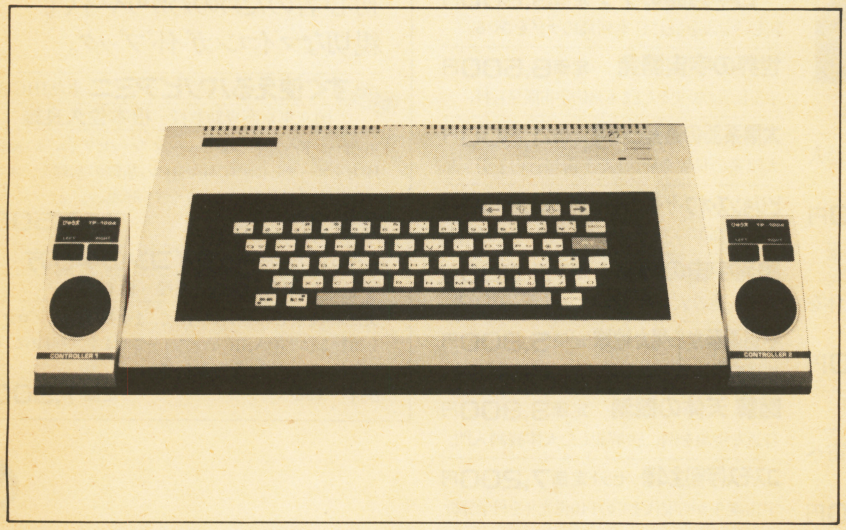


ゲームパソコン（タカラ）
¥59,800

主な特徴

ソードのＭ５と内容的には同等。ただし，こちらの方はゲーム機としてのキャラクターを強めるために，最初からジョイコントローラーを付属しているのが特徴。もちろん，システム拡大をして，プリンター，カセットレコーダーなどを接続すれば，多用途に活用できることはいうまでもない。

●ＣＰＵ：Ｚ80Ａ●ＲＯＭ：８ＫＢ（内蔵モニタ）
●ＲＡＭ：20ＫＢ（内ＶＲＡＭ：16ＫＢ）



ぴゅう太（トミー）
¥59,800

主な特徴

16ビットのゲーム＆ホビー用パソコン。グラフィック機能が充実しているため，ゲームなどの画面がきれいなことが特徴。

ぴゅう太には，いろいろなゲームカートリッジが用意されていて，すぐにでも上質なゲームを楽しむことができるほか，自分自身でオリジナルゲームをかんたんにつくることができるという，プレイ思想が息づいている。

●ＣＰＵ：ＴＭＳ9995●ＲＯＭ：20ＫＢ●ＲＡＭ：16ＫＢ



ＪＲ-200（松下）
本体¥79,800

主な特徴

オーソドックスなパソコンという感じだけど，すべてにアベレージを満足させ，ホームパソコンとして高い充実度をもっている。カラー（８色）機能，３重和音，マシン語モニタを装備するなど，じっくりとつき合っていけば，応用範囲もひろい。

●ＣＰＵ：ＭＮ1800（サブ：ＭＮ1544）●ＲＯＭ：16ＫＢ●ＲＡＭ：32ＫＢ

FROM EDITOR

●マイコンを駆使する高校生が，ソフト作りのアルバイトで得たお金で，親にマンションを買ってあげたという。

●不況の時代でも，コンピュータの知識や技術を持っている人にとっては，就職難という言葉はないといわれる。

●そんなことは百も承知だが，マイコンはマスター不可能と信じている人のためにこそ，本誌は無二の親友となるだろう。

●さて，本誌の創刊に際して最も困難だったのは，ベーシックなゲームとはいえどもどの機種でも楽しめるように，異機種への移植をすることにあった。

●より短時間に，より正確に他機種に移植はできないものか。——この難問を見事に解決してみせたのは，当編集部のマイコン博士K君の抜群のアイデアだった。

●かくして，本誌掲載のゲームプログラムは，多くの機種に打ち込んで楽しむことができるユニークなものとなった。

●《MICOMBOY》第2号は，夏休み前の7月初旬に刊行予定。マイコン好きのあなたと共に，さらに前進していく本誌にご期待ください。　（成瀬）

昭和58年5月1日発行　発行人／赤尾好夫　編集人／石川卓司
発行所㈱旺文社　〒162 東京都新宿区横寺町55
TEL.「編集」(03) 266-6210「販売」(03) 266-6420「広告」(03) 266-6124　定価530円

頭金なしの長期クレジットで超特価!! 今がチャンス

ホームコンピュータ時代到



シャープ MZ731

定価 ¥128,000

超特価22%OFF



神戸三宮9号店

シャープ MZ711

カラー対応 ¥79,800（カラーRF付 テレビ直結可）　価格相談

頭金0円　月々3,900円×24回　　頭金0円　月々3,200円×30回

Ⓐ MZ700＋データーレコーダ＋カラープリンタ（MZ731）¥128,000（ソフト3種付）
頭金0円　月々3,000円×24回　㋭2万×4回
頭金0円　月々4,400円×36回　㋭0万×0回

Ⓑ MZ700＋データレコーダ＋カラープリンタ＋高解カラー2,000字モニタ（ソフト3種付）
定価¥228,000　特価 ~~¥188,000~~
頭金0円　月々4,300円×24回　㋭3万×4回

シャープ MZ2000

MZ2000＋ソフト10種 ¥218,000　価格相談

頭金0円　月々4,100円×24回　㋭4万×4回　　頭金0円　月々4,200円×36回　㋭2万×6回

新登場 CASIO FP1100

Ⓐ FP-1100　¥38,200引

定価¥128,000　27%引

激安目玉品



ウルトラ 特別価格 ¥89,800　ソフト5本付

Ⓑ FP-1100＋高精細カラー＋ソフト5本　定価¥263,000　¥100,000引

ウルトラ 特別価格 ¥163,000
頭金0円　月々3,000円×24回　㋭3万×4回
頭金0円　月々4,000円×36回　㋭1万×6回

Ⓒ FP1100＋高解像シャープカラー¥227,800　激安

ウルトラ 特別価格 ¥197,800　ソフト5本サービス
頭金0円　月々5,400円×48回　㋭　　頭金0円　月々3,500円×36回　㋭2万×6回

apple II J-PLUS

APPLEソフト大量入荷



激安　カラーモニタプレゼント　17%引

Ⓐ J-PLUS＋カラーモニタ付
定価 ~~¥422,800~~ → 大特価 ¥358,000
頭金0円　月々5,800円×36回　㋭4万×6回
頭金0円　月々4,900円×48回　㋭3万×8回

Ⓑ フロッピー（I F付）のみ（ディスケット10枚）
定価 ~~¥199,000~~ → ¥173,000
頭金0円　月々4,300円×36回　㋭1万×6回
頭金0円　月々3,500円×24回　㋭3万×4回

新登場 APPLE II e…¥378,000 ●キーボードが完全アスキー80文字OK!!
64KB RAM（128KB増設可）

シャープRGBカラー



Ⓐ PC9801対応（4,000文字）
PC8801対応（640×400）
大特売 シャープ（ケーブルサービス付）¥158,000
頭金0円　月々3,100円×24回　㋭3万×4回　　頭金0円　月々4,000円×36回　㋭1万×6回

Ⓑ 高解像カラー（640×200）
シャープ（12インチケーブル付2,000文字）¥99,800
頭金0円　月々3,400円×36回　㋭　　頭金0円　月々3,200円×24回　㋭1万×4回

Ⓒ 高精細カラー（2,000文字）グラフィック・漢字対応OK!!
超特価 ¥69,800（ケーブルサービス）

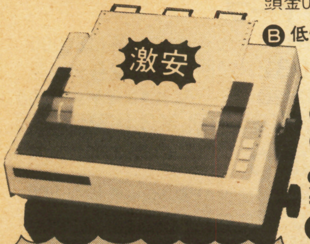
Ⓓ オレンジモニタ（2,000文字）
¥23,800（ケーブルサービス）

Ⓔ グリーンモニタ（2,000文字）
¥24,800（ケーブルサービス）

EPSONプリンター大特価セール!!

プリンタ用紙サービス

Ⓐ 超低価格プリンターRP-80 新発売 ¥89,000
頭金0円　月々2,700円×24回　㋭1万×4回



Ⓑ 低価格高速プリンターFP-80標準機（F101）
新発売 ¥149,800
頭金0円　月々3,500円×36回　㋭1万×6回

●FP-80（PC8001用）¥152,800
●FP-80（PC8801/9801用）¥153,800

Ⓒ MP80K漢字用　¥189,000　プリンタ用紙1,000枚付
¥151,200　頭金0円　月々4,200円×24回　㋭2万×4回

●GPシリーズ大特価セール中!

新製品 SHARPカラープリンター 10インチ巾8色 ¥228,000

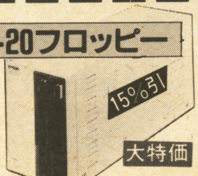
エプソン標準プリンターが漢字プリンターに変身!!
¥4,800（テープ版）　¥6,800（ディスク版）　新漢字ワープロソフト

EPSONフェア!!

4/29㊗～5/1㊐　コンピュータイレブン全店同時開催

EPSON TF-20フロッピー

両面倍密（2W）
560KB～640KB
¥142,000



超低価格フロッピー

3インチフロッピーは小さな巨人!!

新発売 日立 レベル3 用



Ⓐ PC8801対応型　¥142,000＋ファクトリオプション¥24,000 → 特価 ¥142,000
（PC8001対応可）頭金0円　月々4,100円×36回　㋭1万×6回

Ⓑ MICRO-8対応型 ¥142,000＋ファクトリオプション¥21,000 → 特価 ¥139,000
頭金0円　月々4,000円×36回　㋭1万×6回　㊐頭0円　月々4,500円×48回

Ⓒ MZ80対応型 ¥142,000＋ファクトリオプション¥38,000 → 特価 ¥153,000
頭金0円　月々4,500円×36回　㋭1万×6回　㊐頭0円　月々4,900円×48回

SHARP パソコンテレビ X1



定価 ¥155,000　カラーテレビ付モニター¥113,000

システム価格 ¥268,000（ソフト10本サービス）　価格相談

頭金0円　月々9,300円×36回
頭金0円　月々7,400円×48回
頭金0円　月々4,300円×36回　㋭3万×6回　価格相談

新ソフト紹介　ビジネス用簡易言語ソフト
MZ2000・MZ80B用（ディスク版）
今までにない本格的なノンプログラムソフト
データーベース　¥29,800

イレブン新商品情報

- ●GPドットプリンターGP-70……¥98,000
- ●GP漢字プリンターGP-550E……¥118,000
- ●日立マークV……¥特価
- ●SMD 漢字プリンターSX-85K・SX-85K2……¥188,000
- ●MP-80プリンターが漢字プリンターに変身!!「theFM漢字（FM-7/8用）」「thePC漢字（PC8801用）」……¥4,800〈テープ版〉、¥6,800〈ディスク版〉
- ●誰にでもわかるマシン語GAMEの作り方……¥2,800
- ●多重マトリックスデータベース「マルチマット27」（MZ用）……¥29,800
- ●ダスト・カバー（マークII用・FM-7用）……¥1,200
- ●ダスト・カバー（ディスクTF-20用）……¥800
- ●ダスト・カバー（12インチカラーモニタD9用）……¥1,000

資料請求券 ホビー MB-5

㈱日本ソフト&ハード社は教育・開発・出版・卸・小売・通販のパソコン総合企業です。〒171 東京都豊島区高田3-11-14 高田馬場シティハイツ2F ☎03(232)0541代

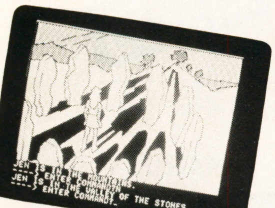
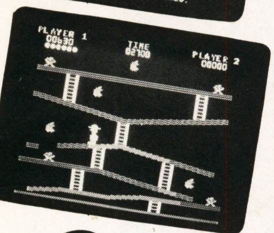
全国に直営ショップを展開中!!

1号店	2号店	3号店	4号店	5号店	6号店	7号店	8号店	9号店	10号店	11号店	12号店
高田馬場	大田梅	名古屋駅前	横浜西口	池袋西口	大阪難波	新宿西口	新橋東口	三宮神戸	梅田ニュー	河原町京都	上野駅前

マイコンショー'83
5/25〜28
東京流通センター展示場

15,000台の販売・サポートのNO.1実績が信用です。

来!!一家に一台パソコンを!

イレブン「プログラムコンテスト」実施中

賞金総額210万円
(特別賞100万円)

●アイデア性のあるゲームソフト・教育ソフト・ビジネスソフト、何でもOK!!
(締切/第2回3月20日・第3回5月20日)

◎特別賞/100万円 ◎一等賞/50万円
◎二等賞/30万円 ◎三等賞/15万円
◎四等賞/5万円 ◎五等賞/1万円

但し、すでに市販のソフト、版権が他人のソフト、現在交渉中のソフトは除きます。連絡先:イレブン開発室☎03(209)5233担当・大沢

「イレブンプログラムコンテスト」第1回〆切分の入選者2名。最終〆切5月20日です。
ふるってご応募下さい。

即決クレジット
ボーナス一括〜60回
日本一安い金利30%off

◎IBMフェアー 4/15▼5/30

◎国産8ビット比較フェアー 4/15▼5/30
シャープMZ731
FM-7
PC8801マークII
大特価

買って安心、後々サポートのコンピュータ
イレブン 全国送料無料保険付

新登場
富士通FM-7
ダストカバー又はソフト7本サービス **¥126,000** 即納

組合せ特別価格 頭金0円 月々2,700円×36回 ㈭1万×6回
(ソフト27種サービス・ケーブルサービス!!)



超低価格のスーパーマイコン

Ⓐ FM-7+グリーンモニタ+ソフト27種
定価¥183,200 特価 **¥159,800**
頭金0円 月々3,900円×36回 ㈭1万×6回
頭金0円 月々2,900円×24回 ㈭2万×6回

Ⓑ FM-7+高精細カラー1,600字+ソフト27種 定価¥236,000 特価 **¥188,000**
頭金0円 月々4,800円×36回 ㈭1万×6回
頭金0円 月々5,200円×48回 ㈭

Ⓒ FM-7+シャープ高解カラー(640×200)+ソフト27種 価格相談
定価¥256,000 特価 **¥214,000**
頭金0円 月々4,100円×36回 ㈭2万×6回



FUJITSU MICRO8
激安35%引 **¥168,000**
MICRO-8+漢字・非漢字SET
¥218,000+¥40,000=**¥258,000**
頭金0円 月々5,600円×36回 ㈭2万×6回
頭金0円 月々5,500円×48回 ㈭1万×8回
頭金0円 月々7,100円×48回
頭金0円 月々6,100円×60回
9万円引

MICRO-8+シャープ高解カラー(640×200)+漢字・非漢字+ソフト7種
¥381,500超特価 **¥278,000** 頭金0円 月々2,700円×48回 ㈭3万×8回

新発売
富士通FM-11 〈AD〉特価¥458,000
¥338,000+シャープ超高解カラー(640×400)
定価¥508,000 頭金0円 月々12,700円×48回
FM-11〈EX〉**¥398,000** 価格相談
大反響

新登場
NEC PC-8001mkII ¥123,000 即納
頭金0円 月々6,000円×24回 頭金0円 月々4,200円×36回
※ソフト7種サービス又はダストカバーサービス

ダストカバー
ソフト7種
サービス

Ⓐ PC8001mkII+シャープ高解カラー(640×200)
定価¥226,300 超特価 **¥209,800**
頭金0円 月々10,400円×24回 頭金0円 月々5,800円×48回
頭金0円 月々 7,300円×36回 頭金0円 月々4,900円×60回

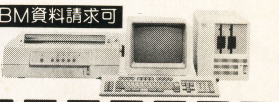
NEC PC8801
Ⓐ PC8801+RFモジュレータ+ソフト27種付 特価 **¥228,000**
頭金0円 月々4,500円×36回 ㈭2万×6回
Ⓑ PC8801+グリーンモニタ+ソフト27種付+レーダーコーダ 特価 **¥248,000**
頭金0円 月々8,900円×24回 ㈭2万×4回 頭金0円 月々8,600円×36回
Ⓒ PC8801+高精細カラー漢字可1600字+ソフト27種付 定価¥331,500 特価 **¥268,000**
Ⓓ PC8801+高解カラーシャープ(640×200)+ソフト27種付 定価¥349,800 特価 **¥298,000**
Ⓔ PC8801+超高解カラーシャープ(640×400)+ソフト27種付 定価¥401,500 特価 **¥368,000**

日本語ビジネス・パソコン(16ビット・256KB標準装備

新登場
IBM IBMマルチステーション 5550

◎最低システムで定価
¥991,000 本体+キーボード+ディスプレイ+フロッピー+プリンタ

★日本語ワード・プロセッサーとして
●作表からグラフまで、いろいろなデータOK
(会計・生産管理・仕入・在庫・予算管理・売掛・売掛・買掛・顧客管理・DM・報告書・伝票発行・名簿・グラフ表示など)
★日本語オンライン端として
●IBM3270漢字情報表示システムと同様の機能
(オンライン業務・オンライン・リアルタイム処理・データベース検索)㊐在庫照会・人事記録照会など

IBM資料請求可


イレブン情報 店頭展示品・新中古品 大特売会〜50%



通販はパソコンプラザ
テレジット
東京 ☎03(209)5266(代)
名古屋 ☎052(451)7374(代)
大阪 ☎06(316)0428(代)

振込先

東京	三菱銀行高田馬場支店㊟4529956コンピュータ・イレブン
池袋	武蔵野銀行池袋支店㊟032433コンピュータ・イレブン池袋西口店
新宿	富士銀行新宿西口支店㊟1558871コンピュータ・イレブン新宿西口店
横浜	平和相互銀行横浜西口支店㊟0306345コンピュータ・イレブン横浜西口店
名古屋	東海銀行笹島支店㊟188805コンピュータ・イレブン名古屋店
梅田	東海銀行梅田支店㊟365820コンピュータ・イレブン大阪梅田店
難波	第一勧業銀行難波支店㊟1003952コンピュータ・イレブン大阪難波店
新橋	大和銀行新橋支店㊟6035336コンピュータ・イレブン新橋東口店
神戸	第一勧業銀行神戸三宮支店㊟1186059コンピュータ・イレブン神戸三宮店
新梅田	太陽神戸大阪駅前支店㊟3124146コンピュータ・イレブンニュー梅田店

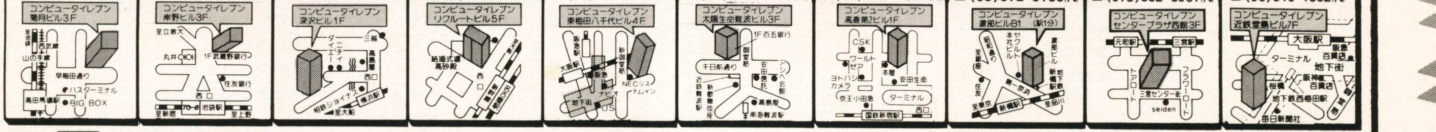
イレブン入学・入社おめでとうセール!! **4月1日▶4月30日お祝いにパソコンを贈ろう!!(超低金利クレジットで楽々)**
5月5日子供の日セール!! 5月1日〜15日

10時〜7時営業
パソコンショップ Computer11
高田馬場駅前店/東京都新宿区高田馬場2-17-4 菊月ビル3F 〒160 ☎(03)209-7376代

池袋西口店	東京都豊島区池袋2-13-1	岸野ビル3F	〒171	☎(03)980-1271代
新宿西口店	東京都新宿区西新宿1-9-13	高倉第2ビル1F	〒160	☎(03)342-4821代
新橋東口店	東京都港区東新橋1-1	渡部ビルB1	〒105	☎(03)572-5166代
横浜西口店	横浜市西区南幸2-5-4	深沢ビル1F	〒220	☎(045)312-4611代
名古屋駅前店	名古屋市中村区椿町1-16リクルート名古屋ビル5F		〒453	☎(052)451-7371代
大阪梅田店	大阪市北区中崎西2-2-1	東梅田八千代ビル4F	〒530	☎(06)316-0546代
大阪難波店	大阪市南区難波2-1-2	太陽生命難波ビル3F	〒542	☎(06)213-7387代
神戸三宮店	神戸市中央区三宮町2-1-5センタープラザ西館3F		〒650	☎(078)332-3961代
大阪ニュー梅田店	大阪市北区堂島2-2-2	近鉄堂島ビル7F	〒530	☎(06)346-1552代

官公庁・学校・企業の方には所定様式で迅速にお見積り致します。
(3年・4年・5年の簡易即決リース有)

東京高田馬場店	東京池袋西口店	横浜西口店	名古屋駅前店	大阪梅田店	大阪難波店	新宿西口店	新橋東口店	神戸三宮店	大阪ニュー梅田店
☎(03)209-7376代	☎(03)980-1271代	☎(045)312-4611代	☎(052)451-7371代	☎(06)316-0546代	☎(06)213-7387代	☎(03)342-4821代	☎(03)572-5166代	☎(078)332-3961代	☎(06)346-1552代



振込先

東京	三菱銀行高田馬場支店㊟4529956コンピュータ・イレブン
池袋	武蔵野銀行池袋支店㊟032433コンピュータ・イレブン池袋西口店
新宿	富士銀行新宿西口支店㊟1558871コンピュータ・イレブン新宿西口店
横浜	平和相互銀行横浜西口支店㊟0306345コンピュータ・イレブン横浜西口店
名古屋	東海銀行笹島支店㊟188805コンピュータ・イレブン名古屋店
大阪	東海銀行梅田支店㊟365820コンピュータ・イレブン大阪梅田店
難波	第一勧業銀行難波支店㊟1003952コンピュータ・イレブン大阪難波店
新橋	大和銀行新橋支店㊟6035336コンピュータ・イレブン新橋東口店
神戸	第一勧業銀行神戸三宮支店㊟1186059コンピュータ・イレブン神戸三宮店
新梅田	太陽神戸大阪駅前支店㊟3124146コンピュータ・イレブンニュー梅田店

クレジットカードOK!! VISA DC UC

レブンフレンドショップ募集!/ マイコンショップの方が☎電話1本で全メーカー商品(本体・周辺・ソフトその他)の仕入が出来るシステム「イレブンフレンドショップ」の加盟店を募集しております。店舗展示・宣伝拡販サポートも行います。お問い合せ☎03(985)6801 担当/高橋・平井

GAME Hudson soft

FM-7

マシン語ソフト続々 新発売

ゲームソフトはハドソン

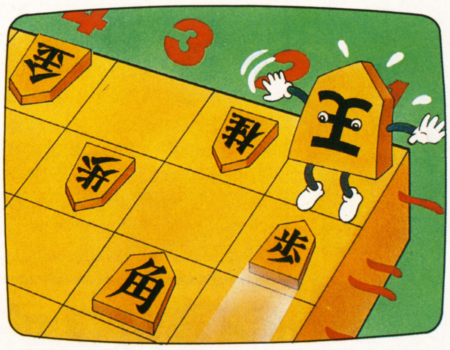
リアルゴルフ

LA-1016 ¥3,800



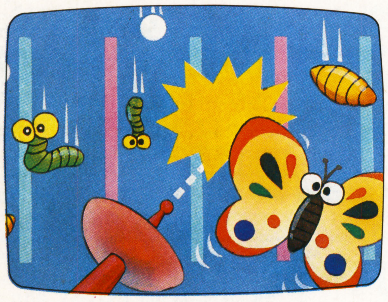
詰将棋

LA-1026 ¥4,800



ミスターバタフライ

LA-1019 ¥3,800



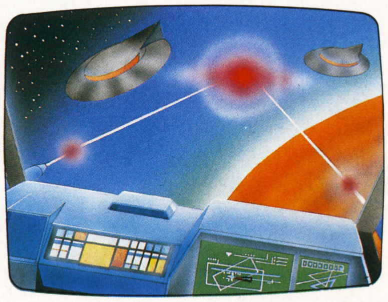
ゾンビパニック

LA-1020 ¥3,800



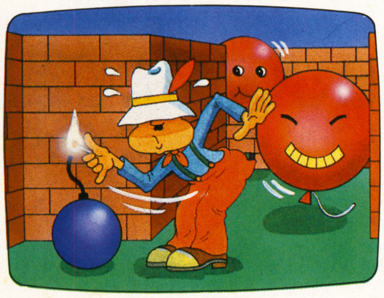
スペーストラブル

LA-1022 ¥3,200



爆弾男

LA-1023 ¥3,200



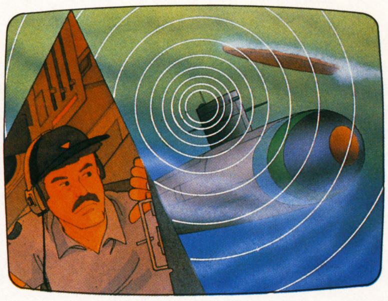
ガンマン

LA-1024 ¥3,800



サブマリンパニック

LA-1025 ¥3,200



オールマシン語新作ソフト&おもしろゲームソフト

ソフト名	SERIAL		言語	PRICE(¥)
夢のプロ野球	LA-1012	巨人VS西武・夢のカードが実現！	B	3,200
リアルゴルフ	LA-1016	全18ホール本格的ゴルフシミュレーション	M	3,800
ミスターバタフライ	LA-1019	あっ！蝶々が襲ってくる	M	3,800
ゾンビ・パニック	LA-1020	ゾンビにパクパク食われないように	M	3,800
野球拳	LA-1021	ちょっと色っぽいおなじみ、よよいのよい	B	3,200
スペーストラブル	LA-1022	熾烈な宇宙戦に勝てるだろうか	M	3,200
爆弾男	LA-1023	風船オバケをやっつけろ！	M	3,200
ガンマン	LA-1024	君は西部のプロガンマン	M	3,800
サブマリンパニック	LA-1025	敵潜水艦は近いぞ！	M	3,200
詰将棋	LA-1026	これで君も将棋のプロになれるぞ	M	4,800

※言語は、Mはマシン語、Bはベーシック。

ソフト名	SERIAL		言語	PRICE(¥)
スタートレック	LA-1001	おなじみのマイコンゲーム古典的名作	B	3,600
アルデバラン#1	LA-1002	レイホープ第二研究所の事故	B	3,600
ビンゴ25	LA-1003	マイコンが勝つか、あなたが勝つか	B	3,800
医は算術なり	LA-1004	あなたは病院を経営できるか	B	3,000
ICBM迎撃作戦	LA-1005	ICBMを地上へ落とすな	B	3,000
サメうちゲーム	LA-1006	あなたはサメ撃ちのスキンダイバー	B	3,000
大戦車突破作戦	LA-1007	地雷をさけて標的を撃て	B	3,000
将棋	LA-1008	ウーン、これはおもしろい	B	5,000
姓名判断	LA-1009	あなたの名前で運勢が決まります	B	4,000
占星術	LA-1010	あなたをすばらしい星占いの世界へ御招待します	B	4,000
家計簿	LA-3001	家庭の主婦のお手伝い	B	3,500

※X1, MZ-700, MZ-2000, FM-7のゲームソフトカタログをご希望の方は、機種を明記の上ハドソン東京まで〒60円を添えて、お申し込み下さい。

※通信販売はソフト名、SERIAL No.を明記の上送料〒300円を添えて、お近くのハドソン0-5係までお送り下さい。
（尚、¥10,000以上お買上げの方は送料無料です。）

GAME

Hudson soft

カシオFP-1100(これもマシン語)こんなソフトをまっていた

アタック・シーモンスター

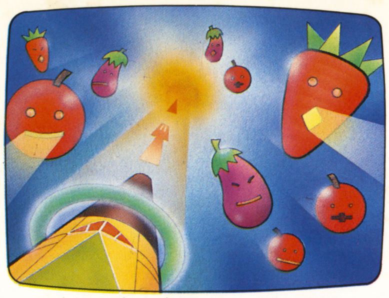
C-1001　言語M　¥3,600



ナント！シーモンスターが、ぞろぞろ襲ってくる！

ベジタブル・クラッシュ

C-1002　言語M　¥3,600



にんじん君、なす君、リンゴ君が登場！

パソコンテレビX1……全てマシン語

クレージースパイ

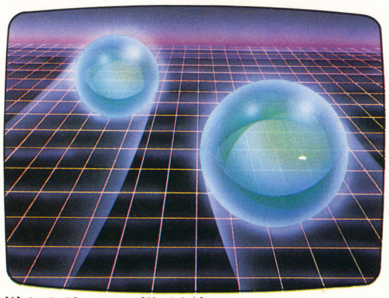
X-1032-G　¥3,200



猛烈な敵宇宙船の攻撃を突破できるか！

キャノンボール

X-1033-G　¥3,800



増えるボールの親玉をキャノンでやっつけろ

あなたは名カメラマン

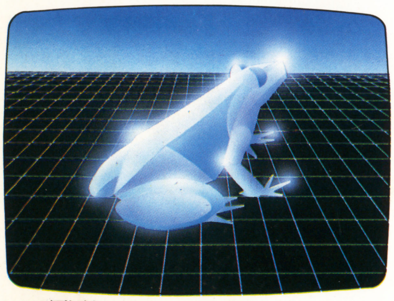
X-1035-G　¥3,200



森の動物写真を撮りましょう

カエルシューター

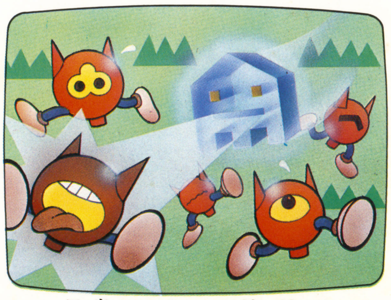
X-1036-G　¥3,800



かえる軍団をオタマジャクシのうちにやっつけろ！

グ　ル　グ　ル

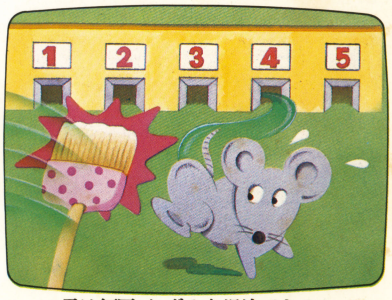
X-1039-G　¥3,200



目・鼻・口のオバケがグルグルと

ねずみとり

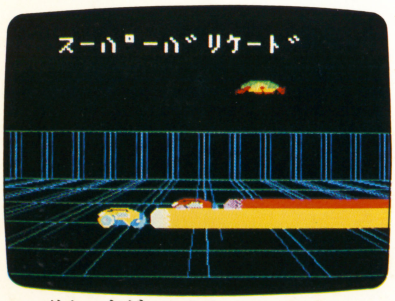
X-1040　¥3,200



君は何匹、ねずみを退治できるか

スーパーバリケード

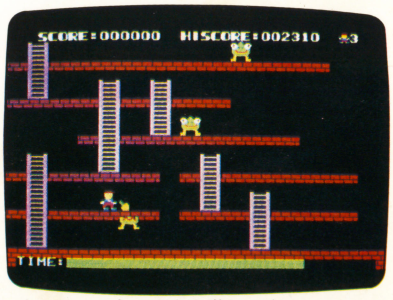
X-1021-G　¥3,200



バリケードゲームのスーパーバージョン

来なさい！

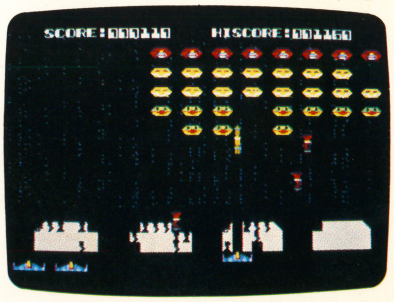
X-1030-G　¥3,800



エイリアンが家の中に大挙して攻めてきました

スペースハンバーガー

X-1001-G　¥3,800



アッ！ハンバーガーが飛んできた。

ゲームソフトはハドソン



パソピアのキーボードに触れる。その瞬間から、もうあなた自身の世界が広がります。パソピアは、付き合う相手を選びません。だから、誰でも気軽に自分なりの自由な世界がつくれます。さあ、あなたも愉快な世界に旅出ちませんか。

多彩なRAM、ROMパックで「パソピア」の機能、さらに充実。

【RAMパック(バッテリーバックアップ付)**】**データ入出力を高速処理でき、電池付きだからデータ保持が可能。あたかも手のひらにのるフロッピーディスクユニットのように手軽に使えます。

- 4KバイトRAM PAC2 PA7240……14,000円
- 16KバイトRAM PAC2 PA7242……28,000円
- 32KバイトRAM PAC2 PA7244……価格未定

※画面のソフトは「コスモトラベラー」T-1601 2,800円



- パーソナルコンピュータ「パソピア」PA7010／7012 163,000円(本体価格)
- ファインカラーディスプレイ PA7161 168,000円
- ミニフロッピーディスク ユニット PA7200 290,000円 OA-DISK BASIC、T-DISK BASICは別売(各18,000円)
- ドットプリンタII PA7251 153,000円

※ケーブルは別売

【漢字ROMパック】ミニフロッピーディスクなしで漢字が使える「漢字ROMパック」新登場。

- 漢字ROM PAC2 PA7246……40,000円

【T-BASIC、OA-BASIC ROMパック】メモリの制約や効率の点で、ひとつのBASICには機能に制限があります。そこで「パソピア」は、用途に応じてBASICを選択できるようT-BASICと、OA-BASICの2タイプをサポートしています。

- T-BASIC ROM PAC1 PA7520……33,000円●OA-BASIC ROM PAC1 PA7522……33,000円●(T-DISK BASIC PA7521、OA-DISK BASIC PA7523……各18,000円)

【MINI-PASCAL ROMパック】標準PASCALのサブセットに独自の機能を持った言語仕様。コンピュータプログラミングの入門用に最適です。スロットインで、BASICマシンからPASCALマシンに早変わり。ミニフロッピーディスクユニットは必要ありません。

- MINI-PASCAL ROM PAC1 PA7540……33,000円

東芝パーソナルコンピュータ
PASOPIA

日本語表示も余裕の16ビット。



●本格的なOAユース、研究開発ユースに対応できる高性能16ビット汎用パソコン。●192Kバイト主メモリに加えて、最大512Kバイトまで拡張可能。●5インチフロッピーディスクユニット、プリンタ、RS-232Cインタフェースを本体に標準実装。●640×500ドットの高解像度。16色指定(ドット単位)やカラーパレット機能で強力なグラフィック機能。●強力な漢字機能に、1000文字(1画面)を表示できる専用ディスプレイ、そして漢字プリンタで日本語処理がいちだんと充実。●薄形軽量のキーボードは、本体とセパレート。Nキーロールオーバーや3段階角度調整機構も装備。●「パソピア」の周辺機器も各種共用可能。

新登場

東芝パーソナルコンピュータ
PASOPIA16

●お問い合わせ、資料ご希望の方は――〒105 東京都港区虎ノ門1-26-5(第17森ビル)東京芝浦電気(株)オフィスオートメーション事業部(03)507-6758・6759(ダイヤルイン)

先端技術をくらしの中に…E&Eの東芝
(エネルギーとエレクトロニクス)

資料請求券 パソピア マイボ

昭和五十八年五月一日発行 発行旺文社 〒一六二 東京都新宿区横寺町五五 電話 東京(03)編集部(二六六)六二一一 販売課(二六六)六四二〇

定価五三〇円

Printed in Japan 雑誌 18307-5

MICOMBOY

NO.1

マイコン入門誌

マイコンボーイ

by モンキー・パンチ

旺文社